2622A/ 2623A

Display Terminals

HEWLETT PACKARD

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

HEWLETT-PACKARD COMPANY
974 EAST ARQUES AVENUE
SUNNYVALE, CALIFORNIA, 94086

Printed in U.S.A. 7/82

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Preface

The HP 2622A Display Terminal is a versatile character, line, or block mode CRT terminal. The HP 2623A Graphics Terminal has the same alphanumeric capabilities as the HP 2622A.

This reference manual contains detailed information for configuring, testing, and using the terminal.

For ease of use, the manual is divided into ten sections and three appendices, as follows:

Section I	General Information
Section II	Configuring the Terminal
Section III	Keyboard Control
Section IV	Display Control
Section V	Printer Control
Section Vi	Data Communications
Section VII	Status
Section VIII	Error Messages and Self-Test
Section IX	Terminal Maintenance Procedures
Section X	Graphics Control Functions
Appendix A	Escape Codes
Appendix B	Keyboards and Character Sets
Appendix C	Programming Examples

Table of Contents

Illustrations

Tables

General Description

SECTION I

INTRODUCTION

The HP 2622A Display Terminal and HP 2623A Graphics Terminal (figures 1-1 and 1-2) are versatile general purpose CRT terminals that offer the following features:

- CRT screen size of 150 mm (6 inches) by 215 mm (8.5 inches).
- Screen capacity of 24 lines of up to 80 characters each (1920 characters total). Two additional lines for system soft key labels.
- Display enhancements (inverse video, blinking, underlining, and half-bright).
- Character, line, or block mode operation.
- Eight user-definable function keys (f1 through f8) are provided. The function of these keys is displayed in two rows of labels across the bottom of the display screen. The labels are definable (up to 16 characters each). The character strings returned when the keys are pressed can be defined with up to 80-character strings.
- All terminal configuration operations are performed through keyboard entries into formatted menus displayed on the screen, there are no physical straps. (Configuration data is maintained in non-volatile memory.)
- Two-character user-definable RETURN key.
- Optional line drawing character set.
- Screen labeled system function keys (for selecting operating modes and performing other terminal control functions).
- Extended Character Set (supports all national keyboard layouts).
- Full editing capabilities (insert/delete/clear line and insert/delete character).
- Adjustable margins and tab stops.
- Programmatic cursor sensing and addressing.
- Easy-to-use keyboard with separate numeric key pad.
- Extensive self-test capability.
- Optional built-in printer—provides either normal, expanded, or compressed character output of alphanumeric data from either the display or datacomm.

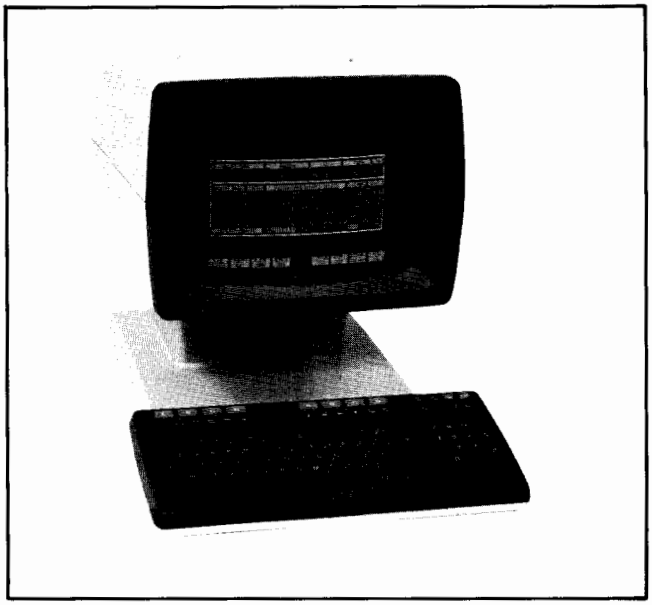


Figure 1-1. HP 2622A Display Terminal

In addition to the HP 2622A features, the HP 2623A Graphics Terminal has the following graphics features:

- Independent graphics and alphanumeric display memories. The graphics memory has 512 by 390 displayable points and selective erase.
- Fast vector generation up to 9600 baud with 11 definable line types.

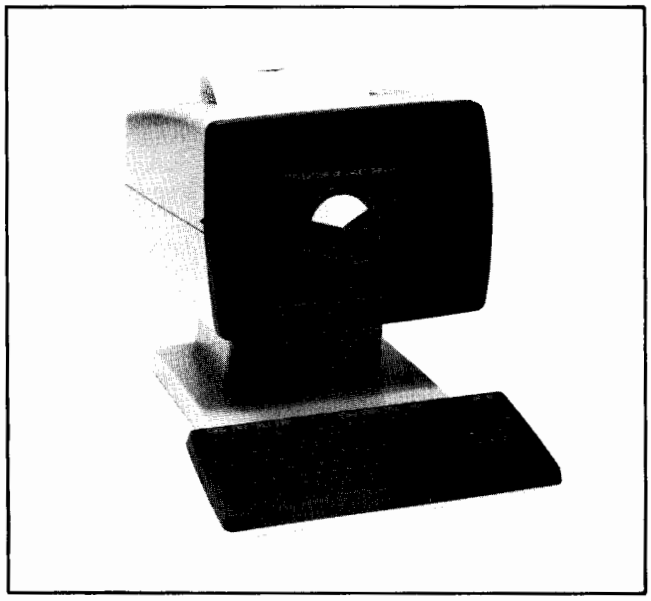


Figure 1-2. HP 2623A Graphics Terminal

- Rectangular area shading with either predefined line types or user-defined pattern.
- Graphics Text Composition—variable character orientation and size; block or slanted; international characters.
- RS232 External Peripheral Port.
- Optional built-in thermal printer—graphics memory hardcopy, 14.5 x 11 cm (5.75 x 4.25 in.) image centered on 8½ inch wide thermal paper, in 60 seconds. Also provides alphanumeric output described for HP 2622A.
- Software support of HP 3000/DSG and HP Graphics 1000/II.
- Compatibility Mode accepts data that has been formatted for Tektronix 4010 and 4012 terminals. *Unscaled* Compatibility Mode displays a 512 by 390 subset of the 1024 by 780 addressable points used in the Tektronix terminals. *Scaled* Compatibility Mode displays a scaled down (512 x 390) version of the 1024 by 780 addressable points.

Table 1-1 lists the specifications for the HP 2622A and the HP 2623A.

Table 1-1. Specifications

GENERAL	
Screen Size:	164 mm (6.5 inches) by 215 mm (8.5 inches)
Screen Capacity:	24 lines of 80 columns (1920 characters). Two additional lines for system soft key labels.
Character Generation:	7 by 11 character in 9 by 15 dot cell
Character Size:	2.4 mm (0.094 inch) by 4.24 mm (0.167 inch)
Character Set:	128 ASCII character set and extended Roman set standard
Cursor:	Blinking-underline
Display Enhancements:	Inverse, underline, blinking, half-bright
Display Modes:	White on black
Refresh Rate:	60 Hz (50 Hz optional)
Memory:	20 K-byte ROM; 3840 bytes (2 pages display memory; 256 bytes of RAM for data communication buffer; 128 bytes of non-volatile RAM (battery backup provided) 512 dots by 390 rows of displayable points (HP 2623A only)
Keyboard:	Detached, with 1.2 m (4 feet) cable. Full ASCII keyboard; 8 screen labeled keys; auto-repeat; N-key rollover. Numeric keyboard and independent user and cursor keys. Combination Numeric Pad and Graphics Control Pad (HP 2623A only).
Operating Modes:	Remote; Local; Character, Line, Block; Forms, Non-Forms.
Transmission Modes:	Full duplex, asynchronous point-to-point. Optional external data communications via modems such as the HP 13265A.
Electrical Interface:	Electrical Industry Association (EIA) Standard RS-232-C
Data Rates:	110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600 baud. Integral printer—120 cps maximum.
Parity:	Selectable: Even, odd, zero, one, none

Table 1-1. Specifications (Continued)

PHYSICAL CHARACTERISTICS	
Weight:	Display Monitor:
	16.8 kg (37 pounds) standard 19.0 kg (42 pounds) with integral printer
	Keyboard:
	2.0 kg (4.4 pounds)
Dimensions:	Display Monitor:
	380 mm wide by 475 mm deep by 440 mm high (15.0 inches by 18.7 inches by 17.3 inches)
	Keyboard:
	430 mm wide by 190 mm deep by 75 mm high (17.0 inches by 7.5 inches by 3.0 inches)
POWER REQUIREMENTS	
Input Voltage:	115 V (+10%, −25%) at 50/60 Hz (±5%) 230 V (+10%, −15%) at 50 Hz (±5%)
Power Consumption:	75 watts (standard) 130 watts (with integral printer)
OPTIONS	
-001	Swedish/Finnish Keyboard
-002	Danish/Norwegian Keyboard
-003	French Keyboard
-004	German Keyboard
-005	United Kingdom Keyboard
-006	Spanish Keyboard
-013	50 Hz, 240 V Power
-014	60 Hz, 100 V Power
-015	50 Hz, 230 V
-016	50 Hz, 115 V
-050	Integral Printer
-201	Line Drawing Character Set

KEYBOARD

The terminal keyboards (see figures 1-3 and 1-4) are divided into five major groups of keys.

Alphanumeric Group—This group of keys is similar to a standard typewriter keyboard and consists of the alphabetic, numeric, and symbol keys. Included are lower and upper case alphabetic characters, ASCII control codes, punctuation characters, and some commercial symbols.

Numeric Pad Group (HP 2622A)—The numeric group of keys is located to the right of the alphanumeric keys. The layout of the numeric key pad is similar to that of a standard office calculator. These keys may be used for high-speed entry of large quantities of numeric data.

Graphics Control/Numeric Pad Group (HP 2623A)—In addition to the numeric pad function, these keys control the graphics display, graphics cursor, and graphics copy to the optional built-in printer or external peripheral. SHIFT NUM toggles the numeric and graphics functions. After a hard reset, CTRL SHIFT RESET, or at power-on, the graphics functions are in effect; the numeric functions are off.

Cursor Control Group—This group of keys is used for moving the cursor around on the screen (up, down, left, or right) and for controlling what portion of the display appears on the screen (home up, home down, roll up, roll down, next page, and previous page).

Edit Control Group—These keys are used for inserting and deleting characters and lines in relation to the current cursor position.

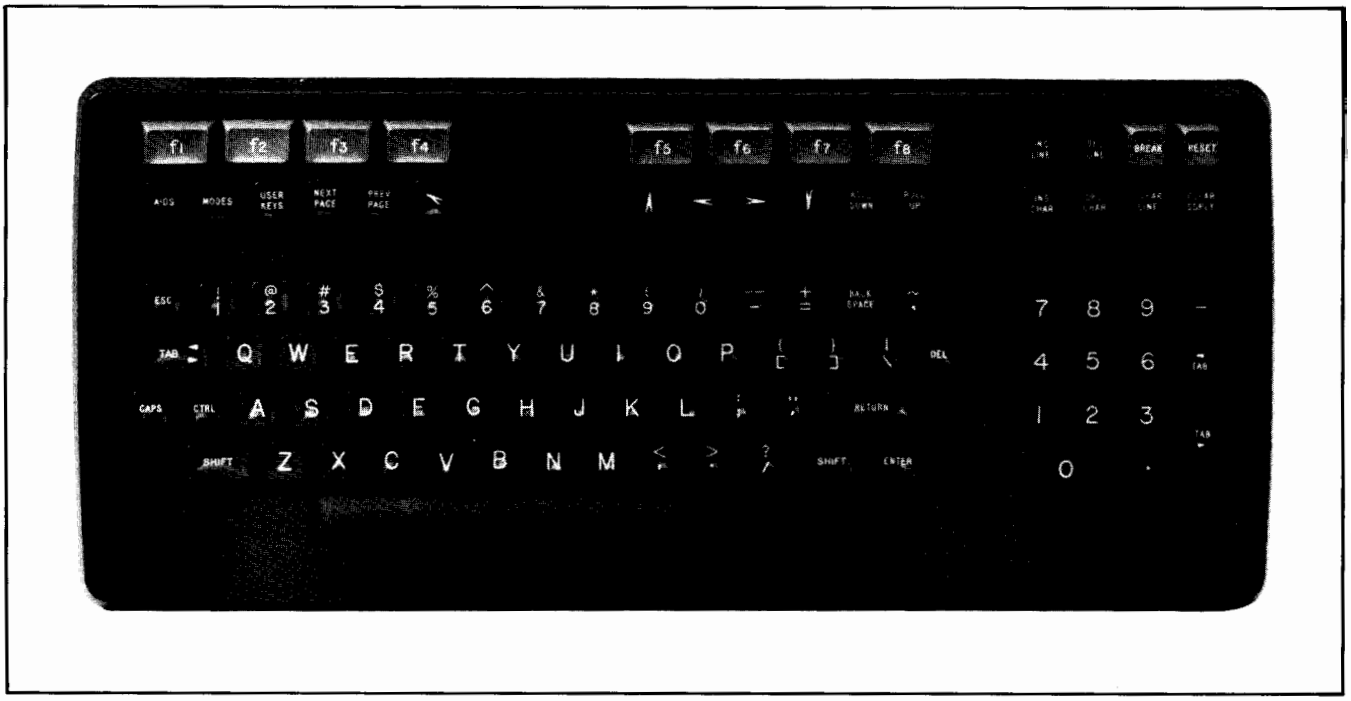


Figure 1-3. HP 2622A Keyboard

Function Group—This group of keys (f1 through f8) perform different functions depending upon which keystrokes have been performed. At any given time the applicable labels for these keys appear across the bottom of the display screen.

The United States (USASCII) keyboard is the standard keyboard. As an option you can order any of the following internationsl keyboards instead:

Swedish/Finnish (Option 001)
Danish/Norwegian (Option 002)
French (Option 003)
German (Option 004)
United Kingdom (Option 005)
Spanish (Option 006)

If you order any of the above keyboards then your terminal automatically includes the optional extended character set

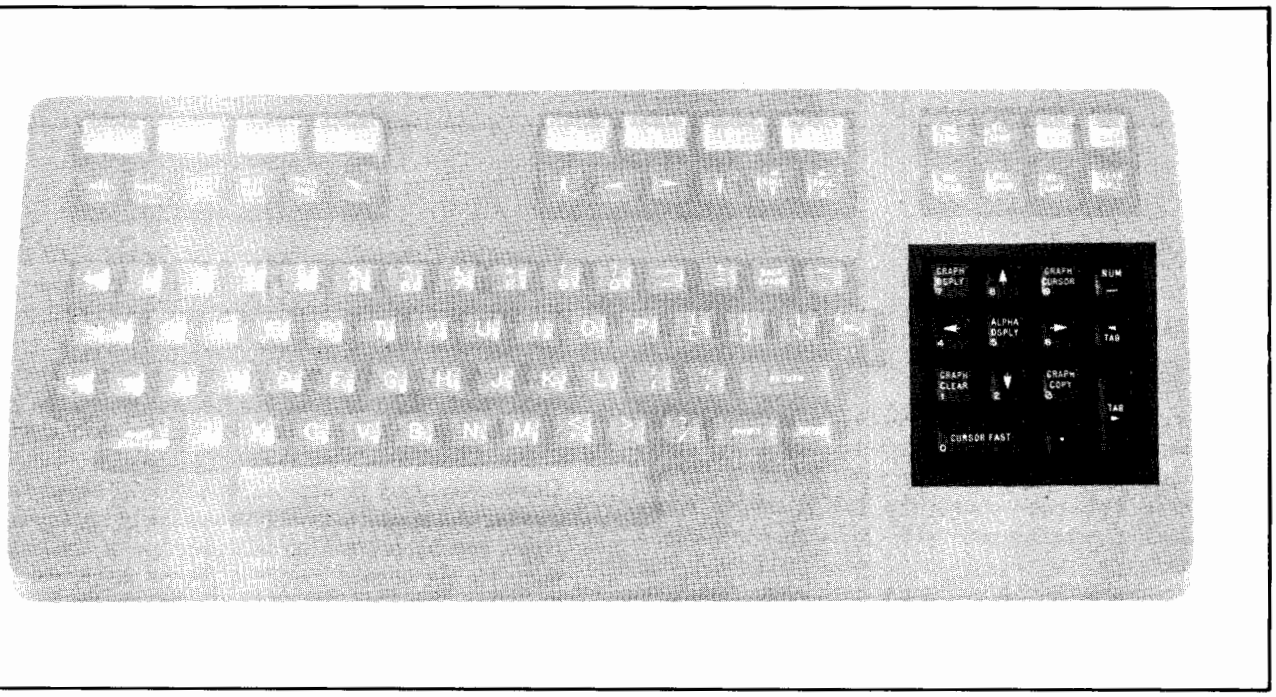


Figure 1-4. HP 2623A Keyboard

(if you order the standard USASCII keyboard, however, then you must explicitly order the extended character set if you want it included). When your terminal includes the extended character set you can select any of the following languages using the configuration process:

USASCII (United States)
Swedish/Finnish
Danish/Norwegian
French AZERTY layout with mute keys
French QWERTY layout with mute keys
French AZERTY layout without mute keys
French QWERTY layout without mute keys
German
United Kingdom
Spanish with mute keys
Spanish without mute keys

FUNCTION KEYS

Across the top of the keyboard are eight keys labeled f1 through f8 (figure 1-5). The functions performed by these keys change dynamically as you use the terminal. At any given time the applicable function labels for these keys appear across the bottom of the display screen.

When you press the MODES key, the eight function keys become mode selection keys (figure 1-6). In this capacity you may use the keys to enable and disable various terminal operating modes (such as remote mode and display functions mode). Each mode selection key alternately enables and disables a particular mode. When the mode is enabled, an asterisk appears in the associated key label on the screen. At power-on, f1 through f8 are automatically initialized as mode selection keys.

Figure 1-5. Function Keys and Screen Labels

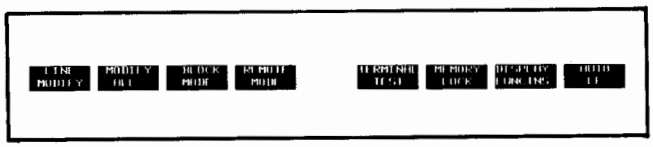


Figure 1-6. Mode Selection Key Labels

When you press the AIDS key, the eight function keys become general control keys that you use for configuring the terminal, setting and clearing margins and tab stops, accessing the Service Keys, and accessing the Device Control group of keys. The entire set of system function key labels for each terminal is illustrated in figure 1-7. Pressing AIDS always reinitializes f1 through f8 to the top row of functions (labels) shown in figure 1-7.

In using the system function keys, keep in mind the following two conventions:

1. If a key label contains any lowercase letters, pressing the key will transfer you to another level of system function keys.
2. If a key label contains only uppercase letters, pressing the key will perform the function defined in the key label.

The key corresponding to `AUTO LF` for example, sets the automatic line feed function on; whereas the key corresponding to `config. keys` transfers you to the configuration function keys.

When you press the SHIFT and USER KEYS keys simultaneously, the user keys definition menu (figure 1-8) appears on the screen. By filling in this menu you can define the screen label and functional characteristics for eight user keys. To enable the eight user keys, press the USER KEYS keys. Figure 1-9 shows the default user key screen labels and figure 1-10 shows some sample user-defined user key screen labels.

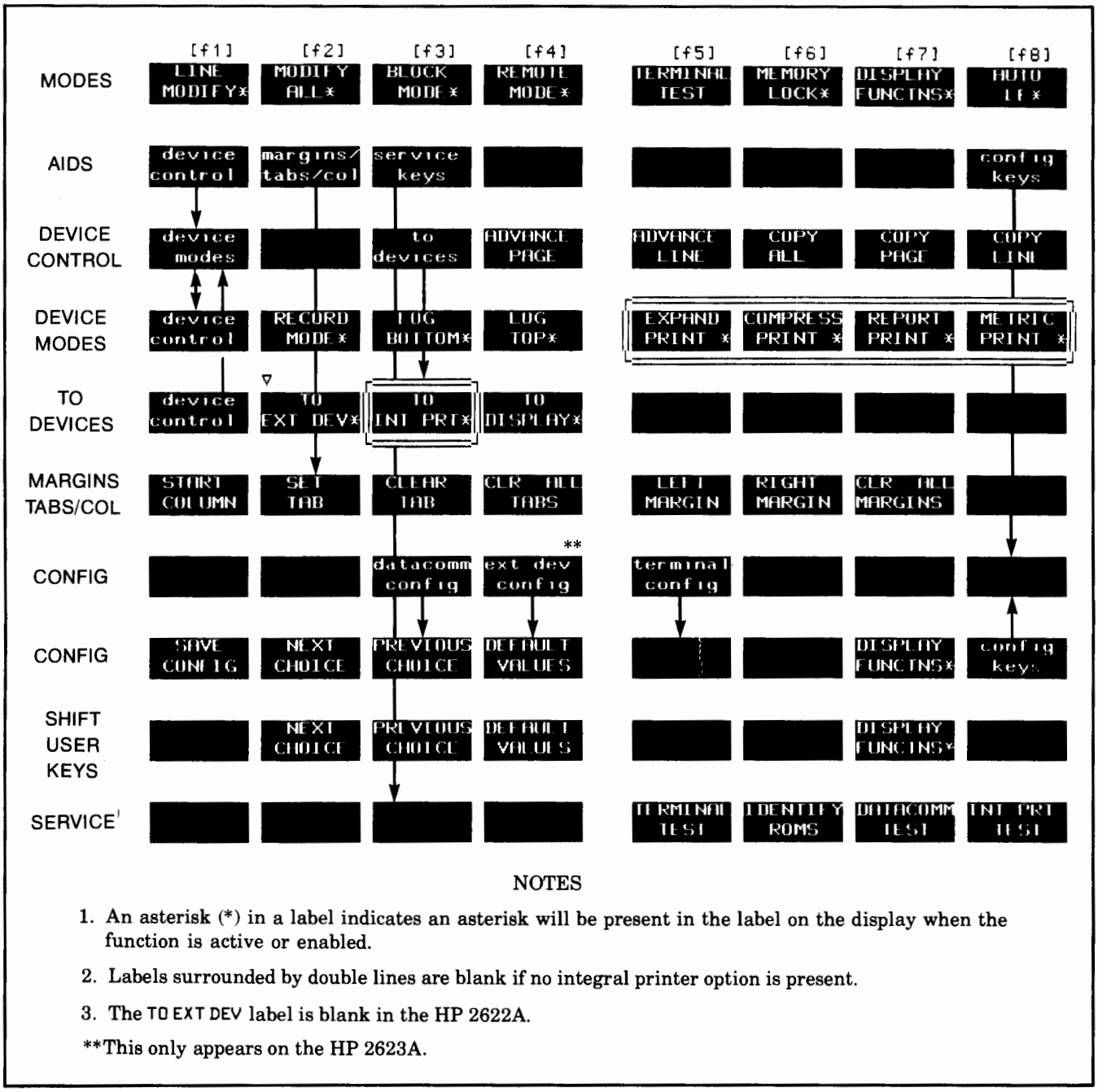
Pressing the SHIFT and AIDS keys removes the key labels from the screen. The User function keys, however, are still enabled. To re-enable the labels, press AIDS, MODES, or USER KEYS.

Refer to Section III for a complete description of the function keys and information on defining these keys.

CONFIGURING THE TERMINAL

The terminal contains no physical straps or switches (other than the ON/OFF switch on the rear panel). You configure the terminal through the use of a configuration menu displayed on the screen.

Refer to Section II for complete information on configuring the terminal.



NOTES

1. An asterisk (*) in a label indicates an asterisk will be present in the label on the display when the function is active or enabled.
2. Labels surrounded by double lines are blank if no integral printer option is present.
3. The TO EXT DEV label is blank in the HP 2622A.

**This only appears on the HP 2623A.

Figure 1-7. HP 2622A and HP 2623A Function Key Hierarchy

INTEGRAL PRINTER

The optional integral printer is a fast, quiet, bidirectional thermal printing unit that can be used for generating reproducible copy from the terminal's display memory or from a remote computer.

The integral printer can reproduce any character capable of being displayed on the terminal's screen, including the line drawing character set and international characters.

Using either the system function keys or escape sequences, you can select the integral printer as the destination device for data transfers. Having done so, you can then perform any of the following types of data transfers using either the system function keys or escape sequences:

- Print the data in compressed, expanded or normal mode.
- Copy one line from the display.
- Copy all lines visible in the display.
- Copy all data from the current cursor line through the end of display memory.

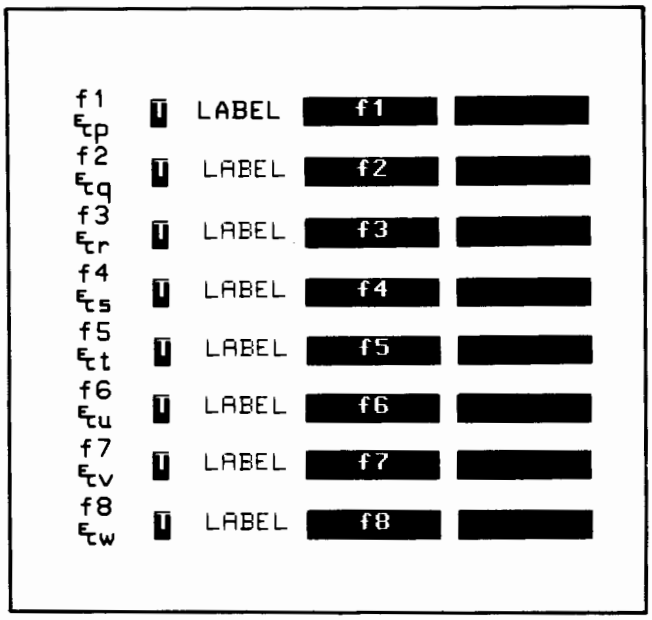


Figure 1-8. User Keys Definition Menu

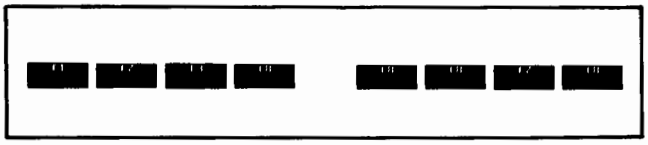


Figure 1-9. Default User Key Labels

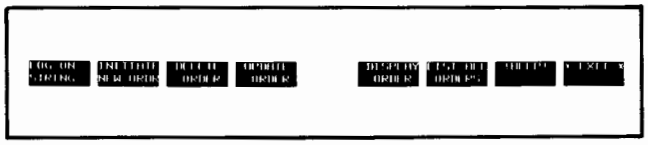


Figure 1-10. Sample User-Supplied User Key Labels

- Copy all of display memory.
- Copy all of graphics memory (HP 2623A only).

In addition, the integral printer offers both report mode and metric mode. When report mode is enabled, the integral printer output is formatted as a series of 8½ inch x 11 inch pages with a top and bottom margin and a tic mark between successive pages. Metric mode produces similar output except that the pages are longer (report mode generates 60 lines of text per page while metric mode generates 64 lines of text per page).

You can also enable data logging to occur from either the top or bottom of display memory. With top logging, any data that is forced off the top of display memory is directed to the integral printer. With bottom logging, any data added to display memory, either from the keyboard or from a datacomm port, is also directed to the printer.

Refer to Section V for additional information on the integral printer.

EXTERNAL PRINTER PORT (HP 2623A Only)

The external printer port at the rear of the HP 2623A terminal provides for interfacing RS-232C printers. The port may be configured to operate from 110 to 9600 baud; choice of 0's, 1's, odd, even, or no parity; 0 to 255 nulls; choice of XON/OFF, SRR, or CS/CB handshaking.

The terminal's electrical interface conforms to EIA RS-232C and CITT V.24 communications interface specifications.

Information on configuring the external printer port is contained in Section V.

DATA COMMUNICATIONS

The terminal can operate at speeds ranging from 110 to 9600 baud.

Transmission can be performed in character mode, block line mode, or block page mode; in all cases the data may be either formatted (a data entry form with unprotected and protected fields) or unformatted.

Using the configuration process, you may enable the following forms of parity generation and checking:

None
Odd
Even
Ones (8th bit forced to 1)
Zeros (8th bit forced to 0)

The terminal's electrical interface adheres to EIA RS-232C and CCITT V.24 communications interface specifications.

Refer to Section VI for complete information on data communications.

SELF-TEST

The terminal is engineered for high reliability, ease of testing, and rapid repair.

When the terminal's power is first turned on, a power-on self-test automatically verifies the integrity of all ROM (Read-Only Memory) and RAM (Random-Access Memory) chips within the terminal. The power-on self-test also does a verification of the configuration data stored in the non-volatile memory.

Using the system function keys, you may also initiate any of the following self-tests:

1. Datacomm Test. This is a very flexible forms-selected test that verifies the integrity of either data communications port. Loop-back via a test hood or a modem may be performed.

2. Terminal Test. This self-test does a CRC verification of all ROM chips within the terminal, non-destructively verifies the integrity of all RAM chips within the terminal, and then displays a test pattern on the screen. The test pattern includes all characters (and segments in the case of the line drawing set) as well as all the character enhancements.

3. Printer Self-Test. This self-test verifies the proper operation of the integral printer. A test pattern containing a variety of characters in standard, compressed, and expanded format is printed.

4. Identify ROMs. This self-test generates a listing (on the display screen) of all installed ROMs.

See Section VIII for complete information concerning the various self-tests.

Configuring the Terminal

SECTION II

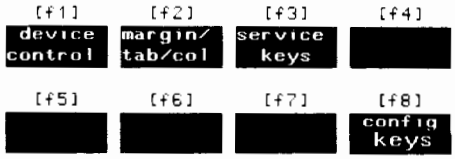
INTRODUCTION

The terminal is designed so that the various terminal characteristics can be configured quickly and easily by displaying configuration "menus" on the screen and then using system function switches to change the content of these menus. Thus altered, the terminal's configuration characteristics are stored in non-volatile memory.

The content of these menus may also be altered from a program executing in a host computer through the use of escape sequences. The changes made by the host computer are temporary and will be lost through hard reset or power down.

CONFIGURATION FUNCTION KEYS

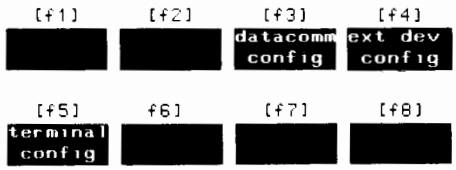
To gain access to the configuration menus through the keyboard, press the shift key. This causes the following label display across the bottom of the screen (f1, f2, etc., refer to the function key corresponding to the label):



NOTE

The device control softkey is present only on terminals equipped with the optional integral printer.

Pressing **config. keys** changes the label display to



Pressing **datacomm config** (f3), **ext dev config** (f4) or **terminal config** (f5) causes a configuration "menu" to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu.

The datacomm menu is described in Section VI, Data Communications. The ext dev config is described in Section V, Printer Control. The terminal configuration menu is described in this section. In addition, information is provided which will enable you to change the terminal configuration menu from the keyboard.

TERMINAL CONFIGURATION MENU

When you press the **terminal config.** (f5) function key, the menu and function key display shown in figure 2-1 or figure 2-2 appear on the screen. Note that the menu as shown in figure 2-1 contains the default settings for all the fields. If you had previously changed the content of any of the fields and then saved the menu in non-volatile memory, the menu would appear on the screen as you had configured it.

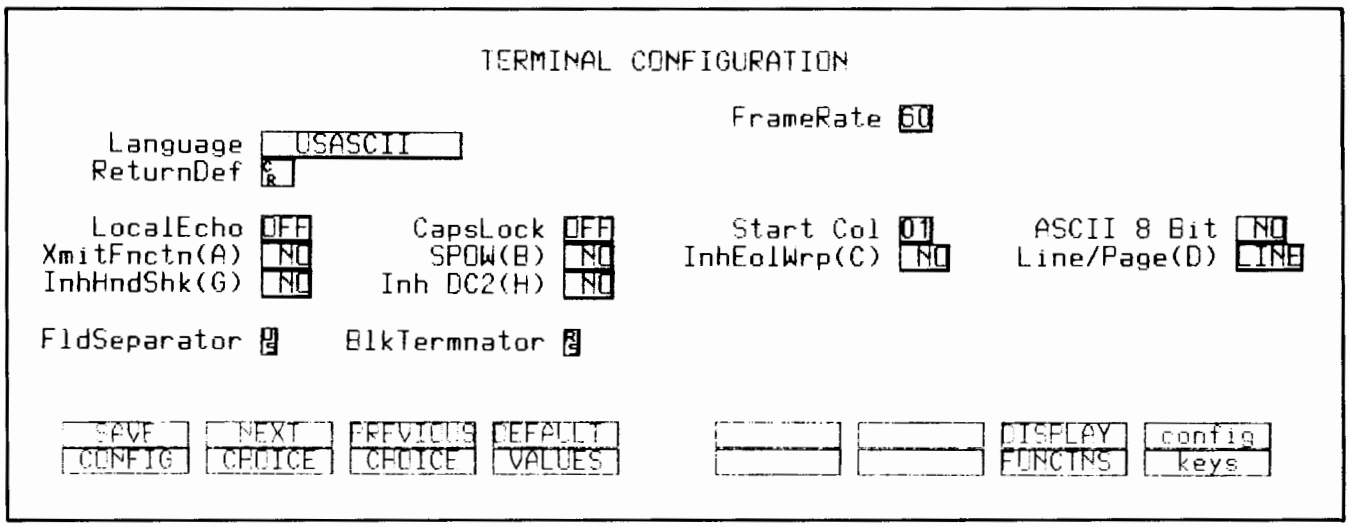


Figure 2-1. HP 2622A Terminal Configuration Menu

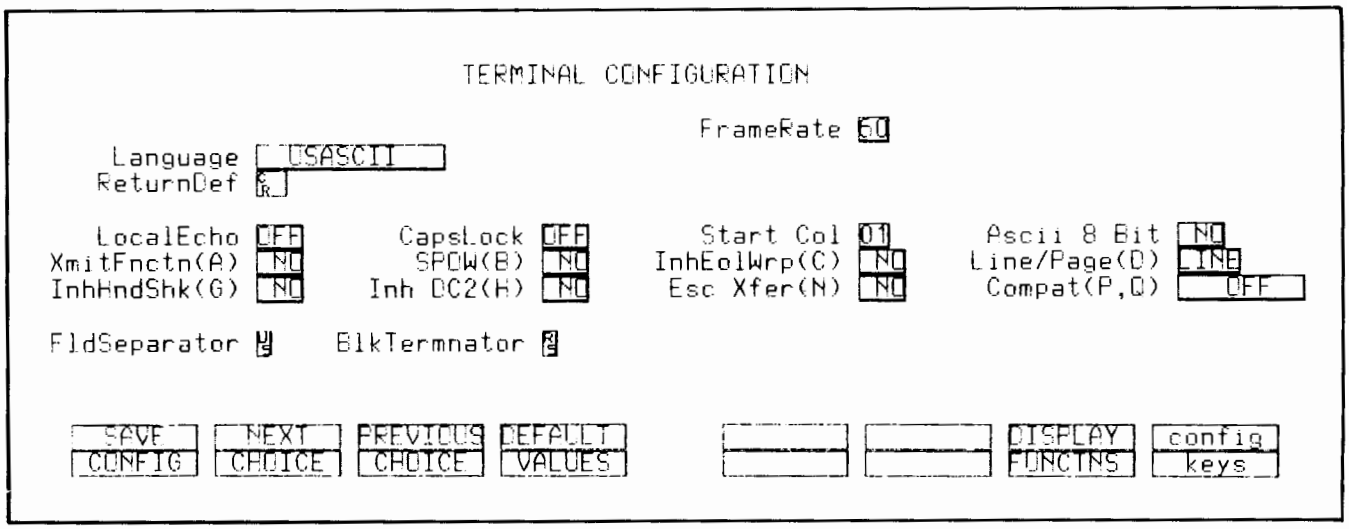


Figure 2-2. HP 2623A Terminal Configuration Menu

NOTE

Whenever a configuration menu is on the screen, no escape sequences are processed, and no incoming or outgoing data is processed, until the menu is exited.

The menu contains a set of unprotected fields that you access using the [TAB▶] and [TAB◀] keys. Except when the cursor is positioned in the fields labeled "Return Def", "FldSeparator", or "BlkTermnator", the alphanumeric keys are disabled and you select the desired parameters using the NEXT CHOICE ([f2]) and PREVIOUS CHOICE ([f3]) function keys.

The meanings of the various fields are described in table 2-1.

Table 2-1. Terminal Configuration Menu Fields

FrameRate	This field specifies what line frequency (50 or 60 Hz) the terminal is designed to operate at. The screen refresh rate is then synchronized to the specified frequency. If this field is set to the wrong value, the images on the screen will pulsate visibly (this will not happen until the save config key is pressed). Values: 50 (for 50 Hz power source) 60 (for 60 Hz power source) The FrameRate can be temporarily selected with the following escape sequences: ESC&k1J = 50 Hz ESC&k0J = 60 Hz Default: 60
RETURN def	This field specifies the definition of the [RETURN] key. The default definition is an ASCII CR. The definition may consist of up to two characters. If the second character is a space, it is ignored. Default: CR
LocalEcho	This field specifies whether characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. ON (ESC&k1L) Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer.

Table 2-1. Terminal Configuration Menu Fields (Continued)

	OFF (ESC&k 0L) Characters entered through the keyboard are transmitted to the host computer only (if they are to appear on the screen, the host computer must "echo" them back to the terminal). Default: OFF
Caps Lock	This field specifies whether the terminal generates the full 128-character ASCII set or only Teletype-compatible codes. ON (ESC&k 1C) The terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents. the {, |, and } keys generate the codes for [, \, and], respectively. The key for generating ~ and ` is disabled. OFF (ESC&k 0C) The terminal generates the full 128-character ASCII set of codes. Default: OFF
Start Col	If the line in which you are entering data is the bottommost used line in display memory (there are no printing or non-printing characters following the current line in display memory), the terminal automatically generates a logical start-of-text pointer to designate the leftmost character that you enter in the line. This pointer remains with the line in display memory until the line is deleted. When you are operating in MODIFY LINE or MODIFYALL mode and you press ENTER or RETURN, the data transmission from the terminal normally begins at the logical start-of-text pointer in the particular line. If the line has no logical start-of-text pointer, however, the data transmission begins at the designated start column. This designated start column can be defined and saved in non-volatile memory using the StartCol field of the terminal configuration menu. The active value of this field can also be temporarily redefined using one of the "margin/tab/col" function keys. Values: 1 - 80 Default: 1
XmitFnctn(A)	This field specifies whether escape code functions are both executed at the terminal and transmitted to the host computer. YES (ESC&s 1A) The escape code sequences generated by control keys such as ROLL UP and ROLL DOWN are transmitted to the host computer. If local echo is ON, the function is also performed locally. NO (ESC&s 0A) The escape code sequences for the major function keys are executed locally but NOT transmitted to the host computer. Note that display functions will emit ESC Z, ESC Y to a host computer. Default: NO
SPOW(B)	This field specifies whether or not spaces entered through the keyboard will overwrite existing characters. NO (ESC&s 0B) Spaces entered through the keyboard will overwrite existing characters. YES (ESC&s 1B) Enable SPace OverWrite (SPOW) latch. When the SPOW latch is off, overwriting occurs as normal. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. Once the SPOW latch has been enabled by the

Table 2-1. Terminal Configuration Menu Fields (Continued)

	above escape sequence, it is turned on by a carriage return and is turned off by a line feed, home up, or tab. It may also be turned on and off programmatically through the use of an ESC&k sequence as follows: ON: ESC&k 1N OFF: ESC&k 0N Default: NO
InhEolWrp(C)	This field specifies whether or not the end-of-line wrap is inhibited. NO (ESC&s 0C) When the cursor reaches the right margin it automatically moves to the left margin in the next lower line (a local carriage return and line feed are generated). YES (ESC&s 1C) When the cursor reaches the right margin it remains in that screen column until an explicit carriage return or other cursor movement function is performed (succeeding characters overwrite the existing character in that screen column). Default: NO
Line/Page(D)	This field specifies whether or not the terminal, when operating in block mode, will transmit data a line at a time or a page at a time. Line (ESC&s 0D) When operating in block mode, the terminal will transmit data a line at a time. Page (ESC&s 1D) When operating in block mode, the terminal will transmit data a page at a time. For a detailed description of the differences between block line and block page mode, refer to "ENTER KEY" in Section III of this manual. Default: LINE
InhHndShk(G) and Inh DC2(H)	Together, these fields determine what type of handshaking is to be used when transferring blocks of data from the terminal to the host computer. The various types of block transfers that may occur are as follows: • A data transfer initiated by pressing the [ENTER] key in character, block line, or block page mode. • A data transfer initiated by pressing the [ENTER] or [RETURN] key in modify mode. • A data transfer initiated by pressing a transmit only (T) user key ([f1] through [f8]). • The terminal's response to a cursor sense, terminal ID status, primary status, secondary status, or device status request issued from the host computer. • The device control completion code (S, F, or U) transmitted by the terminal in conjunction with a device control operation initiated by the host computer. When performing block transfers, there are three possible handshakes: 1. No handshake; terminal merely transmits block of data. 2. Computer sends DC1; terminal transmits block of data. 3. Computer sends DC1; terminal responds with DC2; computer responds with another DC1; terminal transmits block of data.

Table 2-1. Terminal Configuration Menu Fields (Continued)

NOTE

When the ENQ/ACK handshake is enabled and the line/page field of terminal config is selected to be "line", a CR or CR LF is transmitted after the ETX. If "line" is not selected, nothing is transmitted after ETX.

In general, the InhHndShk(G) and Inh DC2(H) fields have the following effects:

InhHndShk(G) = YES	Eliminates the use of the ENQ/ACK handshake (terminal will either use the XON/XOFF/ENQ handshake or no handshake at all).
Inh DC2(H) = YES	Eliminates the use of the XON/XOFF/ENQ handshake (terminal will either use the ENQ/ACK handshake or no handshake at all).
Both = YES	No handshake.

Specifically, however, the type of handshaking used for block transfers is determined by a combination of the following factors:

1. The type of block transfer to be performed.
2. What mode the terminal is currently operating in (character, block line, block page, or modify mode).
3. The setting of the InhHndShk(G) and Inh DC2(H) fields.

If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both NO) are usually adequate for your purposes. If you are concerned about the specific type of handshake to be used for one or more of the particular types of block transfers, however, you should use the following summary to verify (or alter) the settings of the InhHndShk(G) and Inh DC2(H) fields:

1. [redacted] key in block mode; or

   [redacted] or [redacted] key in modify mode; or

   Transmit only (T) user key in block page mode.

   InhHndShk(G) (ignored)

   Inh DC2(H) = NO → XON/XOFF/ENQ

   Inh DC2(H) = YES → no handshake

2. [redacted] key in character mode.

   InhHndShk(G) = YES

   Inh DC2(H) = NO → XON/XOFF/ENQ

   Any other combination → no handshake

3. Transmit only (T) user key in block line or character mode; or

   Cursor sense, terminal ID status, primary status, secondary status, display transfer initiated by ESC d or device status request; or

   Device control completion code.

   InhHndShk(G) = NO → ENQ

   InhHndShk(G) = YES

   Inh DC2(H) = NO → XON/XOFF/ENQ

Table 2-1. Terminal Configuration Menu Fields (Continued)

	InhHndShk(G) = YES Inh␆(H) = YES → no handshake The G and H selections can be temporarily selected by the following escape sequences: ␛&s 0G disables G ␛&s 0H disables H ␛&s 1G enables G ␛&s 1H enables H Defaults: InhHndShk(G) = NO Inh␆(H) = NO
FldSeparator	When you press the [redacted] key while the terminal is in block page mode and display memory contains a formatted display, the terminal automatically transmits the specified field separator character at the end of each protected field (except the final one). Value: Any ASCII character Default: ␤
BlkTermnator	For data transfers between the terminal and a host computer, the terminal (under certain circumstances) transmits the specified block terminator character at the end of the transfer operation. For details, see "The ENTER Key" in Section III. This character, when encountered in display memory, terminates a data transfer ([redacted] key transmissions). Value: Any ASCII character Default: ␍
Language	As will be described in Appendix B, Keyboards and Character Sets, of this manual, the HP 2622A or the HP 2623A can be ordered with any of the following keyboards: United States (standard) Swedish/Finnish (option 001) Danish/Norwegian (option 002) French (option 003) German (option 004) United Kingdom (option 005) Spanish (option 006) With any of the optional keyboards the terminal automatically includes an extended character set that supports the special characters associated with all of the international languages. With the United States keyboard, however, it includes the extended character set only if you have specifically ordered it. When the extended character set is present you may configure the terminal so that the various keys are interpreted (and displayed) in any desired language regardless of which physical keyboard is being used. For example, with a United States keyboard you could configure the terminal so that it responds to the keys as though they were on a German or French or Danish/Norwegian keyboard. This field specifies which national keyboard format is to be used in interpreting keystrokes. Values: USASCII (United States) SVENSK/SUOMI (Swedish/Finnish) FRANCAIS␛zM (French AZERTY layout with mutes)

Table 2-1. Terminal Configuration Menu Fields (Continued)

	FRANCAIS qwM (French QWERTY layout with mutes) FRANCAIS az (French AZERTY layout) FRANCAIS qw (French QWERTY layout) DEUTSCH (German) UK (United Kingdom) ESPANOL M(Spanish with mutes) ESPANOL (Spanish) For the French keyboard layouts, the AZERTY and QWERTY designations refer to the location of the A, Z, Q, and W keys as follows: AZERTY: Row 3 = A Z E R T Y Row 2 = Q S D (etc.) Row 1 = W X C (etc.) QWERTY: Row 3 = Q W E R T Y Row 2 = A S D (etc.) Row 1 = Z X C (etc.) For the French and Spanish keyboard layouts, the mutes designation refers to the manner in which certain accent character keystrokes are handled (' and ¨ on the French layout and ´ on the Spanish). If the mutes are enabled, those keystrokes will generate the particular accent character but will NOT move the cursor. If you then type an applicable vowel, the vowel will appear in the same character position as the accent and the cursor then moves to the next column (if you type any character other than an applicable vowel, however, the character will replace the accent character).
Esc Xfer(N) (HP 2623A only)	This field is the functional equivalent of keyboard interface strap N on an HP 2645 terminal. YES (ESC&s 1N) = When transferring data from display memory to an external printer, escape sequences relating to the display (such as those specifying display enhancements, format mode fields, and alternate character sets) are sent to the external printer if encountered within the data. NO (ESC&s 0N) = Escape sequences relating to the display are not sent to the external printer. NOTE: The Esc Xfer(N) field only affects data transfers between display memory and an external printer. It does NOT affect <Esc>&pW data transfers that go directly from the host computer to the external printer. Default: NO
ASCII 8 Bits	When this operating mode is enabled (=YES), the terminal transmits 8-bit ASCII codes in which the eighth (high-order) bit, when set (=1), indicates that the character is from the alternate character set. This is a Hewlett-Packard convention and you will ordinarily use it only when communicating with certain HP line printers (such as the HP 2635A Printing Terminal). Values: YES (ESC&k 1I) = 8-bit codes. NO (ESC&k 0I) = Standard 7-bit codes. NOTE: When using 8-bit mode, parity should be set to NONE; otherwise, one bit will be used as the parity bit. Also, whenever 8-bit mode is selected, Extended Roman is the default alternate character set; if 7-bit mode is selected, Line Drawing Set is the default alternate character set.
Compat (P,Q) (HP 2623A only)	This field is the functional equivalent of keyboard straps P and Q in the HP 2647A and HP 2648A Graphics Terminals. Either UNSCALED or SCALED entries cause the terminal to enter Compatibility Mode which allows the terminal to respond to graphics control codes for the Tektronix 4010 and 4012 terminals. (Compatibility Mode is explained in detail in Section X.)

Table 2-1. Terminal Configuration Menu Fields (Continued)

	UNSCALED (Esc&s0p 1Q) = Graphics display shows a 512 by 390 subset of the 1024 by 780 addressable points used in the Tektronix terminals. SCALED (Esc&s1p 0Q) = Graphics display shows a scaled down (512 x 390) version of the 1024 x 780 addressable points used in the Tektronix terminals. The entire picture is displayed at approximately ½ the resolution. OFF (Esc&s0p 0Q or Esc&s1p 1Q) = Turns off Compatibility Mode.

Note that as you alter the fields of a configuration menu on the screen, the selected values do NOT alter the content of non-volatile memory nor do they have any effect on the operation of the terminal.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the **save config** ([f1]) function key.

When you do this, the chosen values take effect immediately.

While the terminal configuration menu is displayed on the screen, the [f4], [f5], [f6], [f7], and [f8] function keys perform as shown in table 2-2.

Table 2-2. Configuration Function Keys [f4], [f7], [f8]

[f4] DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values. (For default values, see figure 2-1 or figure 2-2.)
[f7] DISPLAY FUNCTNS	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the Return Def, FldSeparator, or BlkTermnator fields. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the DISPLAY FUNCTNS mode selection key (and vice versa).
[f8] config keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following: [f1] [f2] [f3] datacomm config [f4] ext dev config* [f5] terminal config [f6] [f7] [f8] *ext dev config only on the HP 2623A.

Lock/Unlock Configuration Menus

Using an escape sequence, you can "lock" the current configuration menus (terminal config or datacomm config) so that the menu can not be accessed from the keyboard. Any attempt to access a locked menu from the keyboard will result in a "beep" from the bell. Note that when the configuration menus are locked, the MODIFY ALL (f2), BLOCK MODE (f4), REMOTE MODE (f4), AUTO LF (f8) mode selection keys are also locked.

To lock the menus, use the following escape sequence:

Esc&q1L

To unlock the menus, use the following escape sequence:

Esc&q0L

SETTING CONFIGURATION PARAMETERS WITH ESCAPE CODES

To set the terminal configuration parameters using escape codes, you must use an Esc&k or Esc&s sequence, depending upon which parameters you wish to set.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
FrameRate	Esc&k
LocalEcho	
Caps Lock	
ASCII 8-bit	
SPOW Latch	
XmitFnctn(A)	Esc&s
SPOW(B)	
InhEolWrp(C)	
Line/Page(D)	
InhHndShk(G)	
Inh ℞(H)	
Esc Xfer(N)	(HP 2623A only)
Compat (P,Q)	(HP 2623A only)

The Esc&k and Esc&s sequences alter the particular parameter in the menu, and the new setting takes effect immediately, but they do NOT alter the content of non-volatile memory.

To change the active values of the FrameRate, LocalEcho, Caps Lock, ASCII 8-bit, or SPOW parameters, use an escape sequence of the following form:

```
FrameRate =  60:  Esc&k0J
FrameRate =  50:  Esc&k1J

LocalEcho = OFF:  Esc&k0L
LocalEcho =  ON:  Esc&k1L

Caps Lock = OFF:  Esc&k0C
Caps Lock =  ON:  Esc&k1C

ASCII 8-bit = YES:  Esc&k1I
ASCII 8-bit =  NO:  Esc&k0I

SPOW Latch:  Esc&k0N
             Esc&k1N
```

You may combine these and other Esc&k parameters within one escape sequence. If you do, the final identifier (such as C or I or L) must be uppercase and all preceding identifiers must be lowercase. For example, to set LocalEcho = ON and Caps Lock = ON, you could use either of the following escape sequences:

Esc&k1l1C
Esc&k1c1L

To change the active values of any of the following parameters, use an escape sequence of the following form:

```
XmitFnctn(A) =  NO:  Esc&s0A
XmitFnctn(A) = YES:  Esc&s1A

    SPOW(B) =  NO:  Esc&s0B
    SPOW(B) = YES:  Esc&s1B

InhEolWrp(C) =  NO:  Esc&s0C
InhEolWrp(C) = YES:  Esc&s1C

Line/Page(D) = LINE:  Esc&s0D
Line/Page(D) = PAGE:  Esc&s1D

InhHndShk(G) =  NO:  Esc&s0G
InhHndShk(G) = YES:  Esc&s1G

    Inh ℞(H) =  NO:  Esc&s0H
    Inh ℞(H) = YES:  Esc&s1H
```

(The following two functions apply to the HP 2623A only.)

```
Esc Xfer(N) =  NO:  Esc&s0N
Esc Xfer(N) = YES:  Esc&s1N

Compat (P,Q) =      OFF:  Esc&s0p0Q or Esc&s1p1Q
Compat (P,Q) = UNSCALED:  Esc&s0p1Q
Compat (P,Q) =   SCALED:  Esc&s1p0Q
```

You may combine these and other Esc&s parameters within one escape sequence. If you do, the final identifier (such as A or G or H) must be uppercase and all preceding identifiers must be lowercase. For example, to set Line/Page(D) = PAGE, InhHndShk(G) = NO, and Inh ℞(H) = YES, you could use any of the following escape sequences:

Esc&s1d0g1H
Esc&s0g1h1D
Esc&s1h1d0G

Keyboard Control

SECTION III

INTRODUCTION

The terminal keyboard is a separate unit that is linked to the display portion of the terminal by a flexible cable. Included within the keyboard unit is a speaker that is used for sounding the terminal's bell tone. Except for two keys (RESET and BREAK), the overall keyboard can be logically divided into a character set group, a numeric pad or graphics pad group, a cursor control group, an edit control group, and a function key group. The function key group includes eight keys labeled "f1" through "f8" and the keys labeled "AIDS", "MODES" and "USER KEYS". The f1 through f8 keys are multi-purpose keys in that the functions they perform vary from one situation to another. At any given time the applicable labels for the function keys are displayed across the bottom of the screen (figure 3-1).

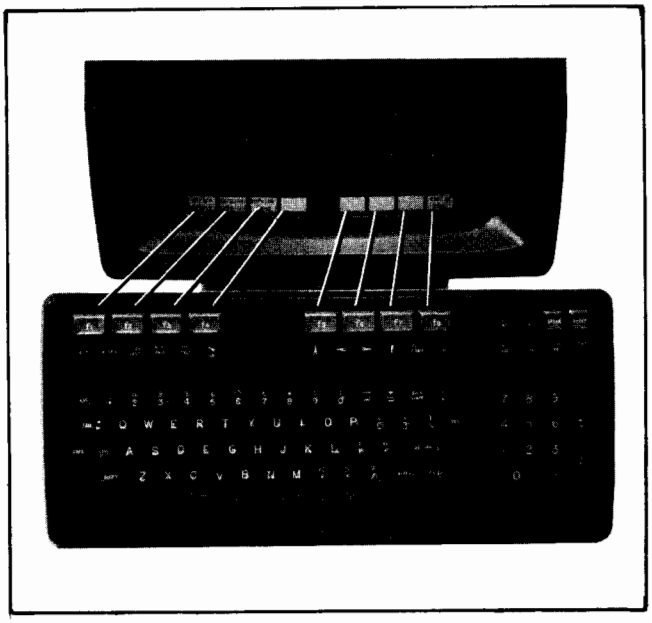
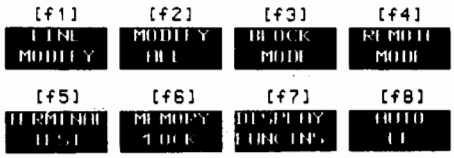


Figure 3-1. Screen-Labeled Function Keys

SELECTING MODES

Pressing the MODES key enables the mode selection keys and changes the f1 through f8 screen labels to the following:



Except for the TERMINAL TEST key (which initiates the terminal self-test), these keys act as toggle switches in that they alternately enable and disable the designated mode. When a particular mode is enabled, an asterisk is displayed in the label.

The graphics functions of the graphics/numeric keypad are in effect when the terminal is powered on or after a hard reset. Also, SHIFT NUM toggles the keypad functions between graphics and numeric functions.

Remote/Local Modes

When a communications link exists between the terminal and a remote host computer, the terminal is in either of the following two modes:

- **Remote Mode.** In this mode, when you press an alphanumeric key the associated ASCII code is transmitted to the host computer.
- **Local Mode.** In this mode, when you press an alphanumeric key the associated character is displayed at the current cursor position on the screen (nothing is transmitted to the host computer).

From the keyboard, you switch the terminal back and forth between local and remote modes using the REMOTE MODE (f1) key.

From the keyboard or a user-definable key, you can switch the terminal from local to remote (and vice versa) using the following escape sequences:

Local: Esc&k 0R
Remote: Esc&k 1R

A remote/local mode designator is maintained in non-volatile memory. When you change modes using the REMOTE MODE key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator in non-volatile memory is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the remote/local designator in non-volatile memory.

Character/Block Modes

When the terminal is connected on-line to a remote host computer, it operates in either of the following data transmission modes:

- **Character Mode.** In this mode, data is transmitted a character at a time as it is entered through the keyboard. ASCII control codes (such as ^A and ^F) are transmitted.

- **Block Mode.** In this mode, data is NOT transmitted at the time it is entered through the keyboard. Instead, you transmit an entire block of data by first typing the data (after initially typing the data you can move the cursor around and edit the data as desired) and then pressing the ENTER key.

When the terminal is in block mode, ASCII control codes (such as ^H and ^F) are acted upon locally but NOT transmitted with the data block.

From the keyboard, you enable and disable block mode using the BLOCK MODE (f3) key.

From a program executing in a host computer, you enable and disable block mode using the following escape sequences:

ENABLE: Esc&k1B
DISABLE: Esc&k0B

A character/block mode designator is maintained in non-volatile memory. When you change modes using the BLOCK MODE key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the character/block designator in non-volatile memory.

The relationship between block, line, page, and format modes is described under ENTER key later in this section.

Format Mode

The terminal includes a format mode in which elaborate, custom-designed forms containing protected and unprotected fields can be displayed on the screen and used for data entry.

When format mode is enabled, the terminal operator may only enter data into unprotected fields. If the operator positions the cursor in a protected area and then attempts to type data, the cursor automatically moves to the start of the next subsequent unprotected field before the terminal accepts the data.

The designing of forms and the use of format mode are described in Section IV.

From a program executing in a host computer or from the keyboard, you enable and disable format mode using the following escape sequences:

ENABLE: EscW
DISABLE: EscX

Once format mode is enabled, it remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Line Modify Mode

When the terminal is in remote mode and character mode, and you are communicating interactively with a host computer, you may sometimes enter an erroneous command string to which the computer responds with an error message. If the command string is a lengthy one and the error consists of only a few characters, it is a nuisance to have to retype the entire string. In such a case, you may instead enable line modify mode (which temporarily switches the terminal to a special form of block mode). You may then move the cursor to the erroneous line on the display and correct the command string. When the string is edited to your satisfaction, you retransmit the line to the host computer by pressing either the RETURN key or the ENTER key.

Note that while line modify mode results in a block transmission, it is completely independent of the block mode function described earlier in this section (you do NOT have to first enable block mode). In fact, line modify mode is a feature that was specifically designed for use when the terminal is operating in character mode.

From the keyboard, you enable line modify mode using the LINE MODIFY key. Line modify mode is automatically disabled when you press either RETURN or ENTER. If you change your mind and wish to disable line modify mode before retransmitting the command string, press the LINE MODIFY key again and the terminal will return to normal character mode.

The terminal remembers which character was the first (leftmost) one that you entered through the keyboard. This means that when you retransmit a line in modify mode, only the keyboard entry portion of the line (the entire edited command string) is retransmitted; any prompt characters preceding the command string are ignored by the terminal. For more detailed information about this feature refer to the discussion of the Start Col field of the terminal configuration menu in Section II.

Modify All Mode

When the terminal is in character mode, you can enable modify all mode, which switches the terminal to a special form of block mode. Modify all mode is the same as line modify mode except that it is NOT disabled when you press RETURN or ENTER.

From the keyboard, you enable and disable modify all mode using the MODIFY ALL key.

From a program executing in a host computer, you enable and disable modify all mode using the following escape sequences:

ENABLE: Esc&k1M
DISABLE: Esc&k0M

A modify all mode designator is maintained in non-volatile memory. When you change modes using the MODIFY ALL key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the modify all designator in non-volatile memory.

NOTE

When using modify mode you will usually want the data block (NOT a P_1 handshake control code) to be sent when you press [ENTER] or [RETURN]. The default configuration parameters, however, enable the $P_1/P_2/P_1$ handshake. Therefore, in most cases you will first need to disable the $P_1/P_2/P_1$ handshake before using modify mode. You do so by setting the Inh DC2(H) field in the terminal configuration menu to "YES", or by sending Esc&s 1H. The modify all and modify line functions are ignored if the terminal is in format or block mode.

Auto Line Feed Mode

When auto line feed mode is enabled, an ASCII line feed control code is automatically appended to each ASCII carriage return control code generated through the keyboard. That is, every CR code generated through the keyboard becomes a CR LF.

ASCII carriage return control codes can be generated through the keyboard in any of the following ways:

- By pressing the [RETURN] key, provided that a CR code is included in the key definition.
- By simultaneously pressing the [CTRL] and [M] keys.
- By pressing any of the user keys ([f1] through [f8]), provided that a CR code is included in the particular key definition.
- By pressing the [ENTER] key when the terminal is in block mode, line modify mode, or modify all mode (in these cases a CR code is transmitted as the line terminator).

From the keyboard, you enable and disable auto line feed mode using the AUTO LF key.

From a program executing in a host computer, you enable and disable auto line feed mode using the following escape sequences:

ENABLE: Esc&k 1A
DISABLE: Esc&k 0A

When you enable or disable auto line feed mode using the "AUTO LF" key, you also alter the content of the "AutoLF" field in both active and non-volatile memory. When you enable or disable auto line feed mode using the escape sequence, however, you only change the content of the "AutoLF" field in active memory.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the "AutoLF" field in non-volatile memory.

Memory Lock Mode

Memory lock mode provides two separate functions: overflow protect and display lock.

OVERFLOW PROTECT. This feature prevents you from losing data when display memory is full. If you home the cursor and then enable memory lock mode, display memory becomes "protected" so that no data can be lost off the top. In such a case, when you have used all available lines in display memory, any attempt to use more memory is rejected with an audible "beep". You may, however, use the cursor control keys to go back and alter any of the existing data. To continue entering new data, merely disable memory lock mode and reposition the cursor immediately below the last line. Before doing so you may wish to enable data logging (described in Section V) so that data that is then forced off the top of display memory will be retained in printed form.

DISPLAY LOCK. If you position the cursor below the top line of the screen and then enable memory lock mode, the lines above the cursor become "locked" on the screen. As the screen becomes full, the locked lines remain on the screen while subsequent lines roll past the locked rows. This allows you to retain column headings or instructions on the screen as you continue to enter new data. It also provides a useful means of changing the sequence of text blocks as follows:

a. Press [↖], [CLEAR DISPLAY], and then type the following data:

```
3. This is paragraph 3. It
   should be the third one.
1. This is paragraph 1. It
   should be the first one.
2. This is paragraph 2. It
   should be the second one.
4. This is paragraph 4. It
   should be the last one.
```

b. Position the cursor in the first line of paragraph 1.
c. Enable memory lock mode.
d. Use the [ROLL UP] key until the first line of paragraph 4 is in the same line as the cursor.
e. Disable memory lock mode and home the cursor. The display should appear as follows:

```
1. This is paragraph 1. It
   should be the first one.
```

2. This is paragraph 2. It should be the second one.
3. This is paragraph 3. It should be the third one.
4. This is paragraph 4. It should be the last one.

From the keyboard, you enable and disable memory lock mode using the MEMORY LOCK key. The rows above the line containing the cursor are locked.

Normal editing can be performed within the locked rows; that is, the rows are locked by row number only, so if lines are inserted among the locked rows, they become locked but the total number of locked rows does not increase.

From a program executing in a host computer, you enable and disable memory lock mode using the following escape sequences:

ENABLE: ESC l
DISABLE: ESC m

Once enabled, memory lock mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Display Functions Mode

When display functions mode is enabled the terminal operates as follows:

- In local mode, it displays ASCII control codes and escape sequences but does not execute them. For example, if you press the ← key the terminal displays ESC D on the screen but does not perform the "cursor left" function.
- In remote mode, it transmits ASCII control codes and escape sequences but does not execute them locally. For example, if you press the ROLL UP key the terminal transmits an ESC S but does not perform the "roll up" function. If local echo is enabled (ON) then the ESC S is also displayed on the screen.

There are two exceptions to the foregoing descriptions:

1. An ESC Z, which disables display functions mode, or ESC Y, which enables display functions mode, is executed but is not transmitted or displayed.
2. A CR (or CR LF if auto line feed mode is enabled) is executed in addition to being transmitted and displayed.

From the keyboard, you enable and disable display functions mode using the DISPLAY FUNCTNS key.

From a program executing in a host computer, you enable and disable display functions mode using the following escape sequences:

ENABLE: ESC Y
DISABLE: ESC Z

NOTE

There is interaction between display functions and the XmitFnctn(A) field of the configuration menu. If XmitFnctn(A) is on, the DISPLAY FUNCTNS key transmits ESC Y, ESC Z. Note that, as a result of this, the receiver of the transmission can never exit display functions.

Once enabled, display functions mode remains enabled until explicitly disabled, until a soft or hard reset is performed, or until the power is turned off.

Caps Mode

When caps mode is enabled, all unshifted alphabetic keys generate uppercase letters and all shifted alphabetic keys generate lowercase letters. This mode is used primarily as a typing convenience and only affects the 26 alphabetic keys.

From the keyboard, you enable and disable caps mode using the CAPS key. This key alternately enables and disables caps mode. The CAPS key has no effect if "Caps Lock" in terminal configuration is enabled (ON).

From a program executing in a host computer, you enable and disable caps mode using the following escape sequences:

ENABLE: ESC & k 1 P
DISABLE: ESC & k 0 P

Once enabled, caps mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Caps Lock Mode

When caps lock mode is enabled, the terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents, the { , | , and } keys generate the codes for [, \ , and] (respectively), and the ` and ~ keys are ignored.

From the keyboard, you enable and disable caps lock mode using the "Caps Lock" field of the terminal configuration menu described in Section II.

From a program executing in a host computer, you enable and disable caps lock mode using the following escape sequences:

ENABLE: ESC & k 1 C
DISABLE: ESC & k 0 C

At any given time the current state (enabled/disabled) of caps lock mode is reflected in the "Caps Lock" field of the terminal configuration menu. When you enable or disable the mode by altering the menu field from the keyboard and then pressing the SAVE CONFIG key, you alter both the active and non-volatile memory versions of that field. When you enable or disable the mode using the escape sequence, however, you only change the active value of the "Caps Lock" field in the terminal configuration menu.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the "Caps Lock" field in the terminal configuration menu in non-volatile memory.

GRAPHICS/NUMERIC KEYPAD ON THE HP 2623A

The graphics functions on the graphics/numeric keypad are enabled when the terminal is powered on, after a hard reset is performed, or when toggled by the SHIFT NUM keys.

- The [GRAPH DSPLY] key toggles the graphics memory display on and off. This allows you to view the alphanumeric memory without the graphics memory overlaying it.
- The [ALPHA DSPLY] key toggles the alphanumeric memory on and off. This allows you to view the graphics memory without the alphanumeric memory overlaying it.
- The [GRAPH CURSOR] key toggles the graphics cursor on and off. Initially, the cursor is positioned in the lower-left corner of the display.
- The ↑, ←, ↓, → keys control the position of the graphics cursor. Two keys may be pressed to cause diagonal movement.
- The [CURSOR FAST] key allows you to position the cursor (using the ↑, ←, ↓, → keys simultaneously) at a more rapid rate.
- The [GRAPH CLEAR] key clears the contents of graphics memory.
- The [GRAPH COPY] key initiates a copy of the contents of graphics memory to the selected "to" devices (either the optional integral printer or the external device connected to the external printer port).

You can toggle the graphics and numeric functions under program control by sending the following escape sequences:

Enable numeric functions: Esc&k0D
Enable graphics functions: Esc&k1D

USER-DEFINABLE KEYS

The eight function keys (f1 through f8), besides performing their usual terminal control functions, can be defined either locally by the terminal operator or remotely by a program executing in a host computer. By "defined" it is meant that:

1. You can assign to each key a string of ASCII alphanumeric characters and/or control codes (such as ^R or ^F).
2. You can specify each key's operational attribute: whether its content is to be executed locally at the terminal, transmitted to a host computer, or both.
3. You can assign to each key an alphanumeric label (up to 16 characters) which, in user keys mode, is displayed across the bottom of the screen.

When defining a key from the keyboard, the key content may include explicit escape sequences (entered using display functions mode) that control or modify the terminal's operation.

The definition of each user key may contain up to 80 characters (alphanumeric characters, ASCII control characters, and explicit escape sequence characters).

Defining Keys Locally

To define one or more keys from the keyboard, first press the SHIFT and USER KEYS keys simultaneously, or use Escj . The user keys menu shown in figure 3-2 then appears on the screen. Note that the menu in figure 3-2 contains the default values for all of the fields. While the menu is displayed on the screen, you can reset the entire menu to the default values by pressing the DEFAULT VALUES function key (f4).

The menu contains a set of unprotected fields that you access using the TAB▶ and TAB◀ keys.

For each user key the menu contains four unprotected fields:

ATTRIBUTE FIELD. This one-character field always contains an uppercase L, T, or N signifying whether the content of the particular user key is to be:

a. Executed locally only (L).
b. Transmitted to the host computer only (T).
c. Treated in the same manner as the alphanumeric keys (N). If the terminal is in local mode, the content of the key is executed locally. If the terminal is in remote mode and local echo is disabled (OFF), the content of the key is transmitted to the host computer. If the terminal is in remote mode and local echo is enabled (ON), the content of the key is both transmitted to the host computer and executed locally.

The alphanumeric keys are disabled when the cursor is positioned in this field. You change the content of this

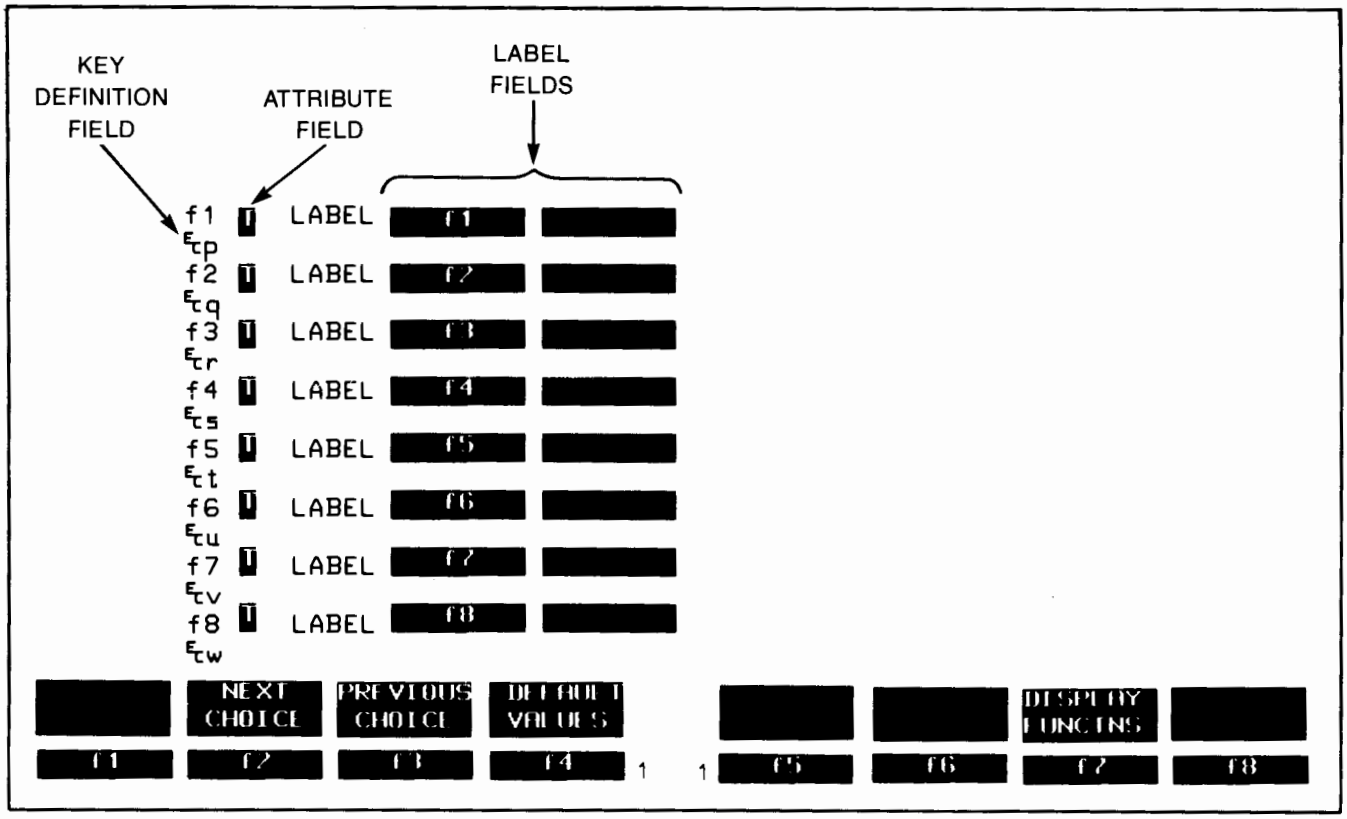


Figure 3-2. User Keys Definition Menu and Default User Key Labels

field by pressing the NEXT CHOICE and PREVIOUS CHOICE keys (f2 and f3, respectively).

TWO LABEL FIELDS. The two eight-character fields to the right of the word "LABEL" allow you to supply the user key's label. When the terminal is in user keys mode, the key labels are displayed from left to right in ascending order across the bottom of the screen (each displayed key label occupies two lines). The first LABEL field in the user keys menu supplies the upper portion of the particular key label while the second supplies the lower portion.

KEY DEFINITION FIELD. The entire line (80 characters) immediately below the attribute and label fields is available for specifying the character string that is to be displayed, executed, and/or transmitted whenever the particular key is physically pressed. When entering characters into this field you may use display functions mode.

When entering the label and key definition you may access display functions mode by way of the DISPLAY FUNCTNS function key (f7). Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys.

The RETURN can be used for including ^M codes (with display functions mode enabled) in key definitions. If auto line feed mode is also enabled, the RETURN key will generate a ^M ^J, otherwise it is considered a cursor movement key.

When the user keys menu is displayed on the screen you may use the INSERT CHAR, DELETE CHAR, and CLEAR LINE keys for editing the content of the label and key definition fields.

When you are finished defining all the desired keys, press the AIDS, MODES, or USER KEYS key (in all three cases the user keys menu disappears from the screen). When you press USER KEYS, or enter Ek, the defined user key labels are displayed across the bottom of the screen and the f1 through f8 user keys, as defined by you, are enabled.

Defining Keys Programmatically

From a program executing in a host computer, you can define one or more keys using the following escape sequence format:

```
Ec&f          <attribute><key><label length>
              <string length><label><string>
```

where:

<attribute>	=	0a: normal 1a: local only 2a: transmit only	(0 is the default)
<key>	=	1-8k: f1-f8, respectively	(1 is the default)
<label length>	=	0-16d	(0 is the default)
<string length>	=	0-80L	(1 is the default) (-1 causes field to be erased)

<label> = the character sequence for the label field
<string> = the character sequence for the key definition field

The <attribute>, <key>, <label length>, and <string length> parameters may appear in any sequence but must precede the label and key definition strings. You must use an uppercase identifier (A, K, D, or L) for the final parameter and a lowercase identifier (a, k, d, or l) for all preceding parameters. Following the parameters, the first 0 through 16 characters, as designated by <label length>, constitute the key's label and the next 0 through 80 characters, as designated by <string length>, constitute the key's definition string. The total number of displayable characters (alphanumeric data, ASCII control codes such as ↖ and ↳F, and explicit escape sequence characters) in the label string must not exceed 16, and in the definition string must not exceed 80.

Example: Assign LOG-ON as the label and HELLO USER.ACCOUNT as the definition for the f5 user key. The key is to have the attribute "N".

```
&f5k6d19LLOG-ONHELLO USER.ACCOUNT↖&jB
```

After issuing the foregoing escape sequence from your program to the terminal, the f5 portion of the user keys menu is as follows:

```
f5 N LABEL LOG-ON
HELLO USER.ACCOUNT↖
```

If the transmit only attribute (2) is designated, the particular user key will have no effect unless the terminal is in remote mode. A transmit only user key may (when subsequently pressed) invoke a block transfer handshake and append the appropriate terminator to the string. The &jB sequence turns on the user labels. If this escape sequence is not sent, the labels will not be updated unless the USER KEYS key is pressed twice.

Controlling the User Keys Menu Programmatically

From a program executing in a host computer, you can display the user keys menu on the screen and remove it from the screen using the following escape sequences:

DISPLAY MENU: &j
REMOVE MENU: &k

Controlling the Function Key Labels Programmatically

From a program executing in a host computer, you can control the function key labels display as follows by using escape sequences:

- You can remove the key labels from the screen entirely (this is the equivalent of simultaneously pressing the SHIFT and CTRL keys).
- You can enable the mode selection keys (this is the equivalent of pressing the MODES key).
- You can enable the user keys (this is the equivalent of pressing the USER KEYS key).
- You can "lock" the current set of labels on the screen (i.e., disable the CTRL, MODES, and USER KEYS keys).
- You can reenable the CTRL, MODES, and USER KEYS keys.

The escape sequences are as follows:

&j@	Enable the user keys and remove all key labels from the screen.
&jA	Enable the mode selection keys.
&jB	Enable the user keys.
&jR	Unlock screen labels.
&jS	Lock screen labels.
&j<xx>L<message>	Remove the key labels from the screen and display the character string <message> (which consists of <xx> characters; where <xx> may be up to 160 characters).
&jC	Remove your <message> from the screen and restore the current key labels.

ENTER KEY

When the terminal is in remote mode, pressing the ENTER key sets pending a block transfer of data from display memory to the host computer (in such a case the ENTER key also locks the keyboard until the resultant data transfer is complete).

The type of handshaking used and precisely what data gets transmitted depends on the following factors:

1. Whether the terminal is in character mode, block line mode, or block page mode.
2. Whether or not the terminal is in format mode.
3. The settings of the InhHndShk(G), Inh DC2(H), and line/page fields in the terminal configuration menu.

Table 3-1 summarizes the effect of the ENTER key in each of the possible mode/strap combinations.

Table 3-1. [redacted] Key Operation

CHARACTER MODE

The cursor is repositioned to column 1.

All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a ↵ (or a ↵ LF if auto line feed mode is enabled). The cursor is repositioned to column 1 and a line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a ↵ (or a ↵ LF if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a ↵ (or a ↵ LF if auto line feed mode is enabled).

The type of handshaking used is determined as follows:

InhHndShk(G) = YES

Inh DC2(H) = NO → $P_1/P_2/P_3$

Any other combination → no handshake

CHARACTER MODE, FORMAT MODE

If the cursor is within an unprotected field, all characters from the current cursor position through the end of the field are transmitted to the host computer as a block. Otherwise, the terminal searches for the next subsequent unprotected field and transmits the content of that field.

ASCII control codes within the field are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a ↵ (or a ↵ LF if auto line feed mode is enabled). The cursor remains at the first character position after the end of the field.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a ↵ (or a ↵ LF if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a ↵ (or a ↵ LF if auto line feed mode is enabled). The ↵ that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determined as follows:

InhHndShk(G) = YES

Inh DC2(H) = NO → $P_1/P_2/P_3$

Any other combination → no handshake

BLOCK LINE MODE

Block line mode means that Block Mode is on and that the line/page(D) field in terminal configuration is set to "line".

Inh DC2(H) = YES

The cursor is repositioned to one within the current line. All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are then transmitted to the host computer as a block.

Inh DC2(H) = NO

The cursor is NOT repositioned. All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a ↵ (or a ↵ LF if auto line feed mode is enabled). The cursor is repositioned to column 0 and a line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a ↵ (or a ↵ LF if auto line feed mode is enabled).

The cursor remains positioned immediately following the terminator.

Table 3-1. [redacted] Key Operation (Continued)

If there is no data to transmit, a block terminator followed by a ↵ or ↵ LF is transmitted. The cursor is not moved.

The type of handshaking used is determined as follows:

InhHndShk(G) is ignored

Inh DC2(H) = NO → $D_1/D_2/D_1$

Inh DC2(H) = YES → no handshake

BLOCK LINE MODE, FORMAT MODE

Block line mode means that Block Mode is on and the line/page (D) field in terminal configuration is set to "line".

ASCII control codes within the field are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a ↵ (or a ↵ LF if auto line feed mode is enabled). The cursor remains positioned at the end of the field.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a ↵ (or a ↵ LF if auto line feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a ↵ (or a ↵ LF if auto line feed mode is enabled). The ↵ that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determined as follows:

InhHndShk(G) (ignored)

Inh DC2(H) = NO → $D_1/D_2/D_1$

Inh DC2(H) = YES → no handshake

BLOCK PAGE MODE

Block page mode means that Block Mode is on and the line/page (D) field in terminal configuration is set to "line".

Inh DC2(H) = YES

The cursor is repositioned to the "home up" position. All characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one line in display memory.

Inh DC2(H) = NO

The cursor is NOT repositioned. All characters from the cursor position through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one line in display memory.

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

After each line (except the final one) the terminal sends a ↵ LF. If the operation is terminated by encountering the end of display memory, the terminal sends a ↵ LF followed by a block terminator after the last line. If the operation is terminated by encountering a block terminator, the terminal sends only a block terminator after the last line.

If there is no data to be transmitted, the terminal sends only a block terminator.

The type of handshaking used is determined as follows:

InhHndShk(G) (ignored)

Inh DC2(H) = NO → $D_1/D_2/D_1$

Inh DC2(H) = YES → no handshake

BLOCK PAGE MODE, FORMAT MODE

Inh DC2(H) = YES

The cursor is repositioned to the "home up" position. All unprotected characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one unprotected field.

Inh DC2(H) = NO

The cursor is NOT repositioned. All unprotected characters through the first subsequent block terminator or through the end of display memory

Table 3-1. [ENTER] Key Operation (Continued)

(whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one unprotected field.

ASCII control codes within the fields are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the fields are NOT transmitted.

After each field (except the final one), the terminal sends a field separator. After the final field, the terminal sends a block terminator.

If the end of display memory is encountered before locating an unprotected field, the terminal merely sends a block terminator.

The type of handshaking used is determined as follows:

InhHndShk(G) (ignored)

Inh DC2(H) = NO → $P_1/P_2/P_1$

Inh DC2(H) = YES → no handshake

For more information on the logical start-of-text pointer and start column, refer to Table 2-1, "Terminal Configuration Menu Fields" in section II.

All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are transmitted to the host computer as a block.

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a CR (or a CR LF if auto line feed mode is enabled). If LocalEcho = OFF, the cursor is repositioned to the column at which the transmission began, otherwise the cursor is repositioned to Column 1. A line feed is performed if auto line feed mode is enabled.

MODIFY MODE

Note that modify line and modify all modes are functional only when the terminal is configured for character mode operation. When either block mode or format mode is enabled, the [ENTER] key operates as described for block mode earlier in this table.

In modify mode, the cursor is repositioned as follows:

1. To the logical start-of-text pointer; or
2. To the designated start column (Start Col) if there is no logical start-of-text pointer.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled).

The cursor remains positioned immediately following the terminator.

The type of handshaking used is determined as follows:

InhHndShk(G) is ignored

Inh DC2(H) = NO → $P_1/P_2/P_1$

Inh DC2(H) = YES → no handshake

SEND DISPLAY (Esc d)

From a program executing in a host computer, you can trigger a block transfer of data from display memory to the host computer by issuing the following escape sequence:

Esc d

This escape sequence is only responded to when received over a datacomm line; it is ignored if entered through the keyboard or issued from a user key (unless block mode is enabled). With the following two exceptions, the resultant data transfer is performed as though the [ENTER] key had been pressed:

1. The cursor is NOT repositioned. The data transfer always begins at the current cursor position.
2. The type of handshaking used is determined as follows:

   InhHndShk(G) = NO → P_1 handshake

   InhHndShk(G) = YES

   Inh DC2(H) = NO → $P_1/P_2/P_1$

   InhHndShk(G) = YES

   Inh DC2(H) = YES → no handshake

The Ectd sequence also temporarily disables the keyboard so that the ENTER key cannot be used until the current data transfer is completed. If the Ectd sequence is received while an ENTER key data transfer is in progress, the escape sequence is ignored.

Note that an Ectd sequence resets the "block trigger received" flag. This means, for example, that if you are using the ^Q handshake and the terminal receives a ^Q followed by the Ectd, it "forgets" that a block trigger was just received and thus will NOT send the data immediately. The terminal must receive another ^Q before it will start the data transfer.

The amount of data transferred depends on the following terminal settings: page/line field in terminal configuration, and whether Block Mode, Modify All, or Line Modify is enabled. For more detailed information, refer to Table 3-1 "ENTER Key Operation" in this section.

ENABLE/DISABLE KEYBOARD

You can enable and disable the terminal's keyboard by executing escape sequences. When the keyboard is disabled all keys EXCEPT the following are ignored:

SHIFT CTRL RESET BREAK

The escape sequences for enabling and disabling the keyboard are as follows:

ENABLE: Etb
DISABLE: Etc

Once disabled, the keyboard remains disabled until explicitly enabled, until a soft or hard reset is performed, or until the power is turned off.

SOFT RESET

A soft reset does the following:

1. Rings the terminal's bell.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
5. Disables display functions mode (if enabled).
6. Halts any datacomm transfers currently in progress, clears the datacomm buffers.
7. Resets the integral printer, if present.
8. Turns off record mode, if on.

The data on the screen, all terminal operating modes (except display functions mode), and all active configuration parameters are unchanged.

From the keyboard, you perform a soft reset by pressing the RESET key.

From a program executing in a host computer, you perform a soft reset using the following escape sequence:

Etg

HARD RESET

A hard reset has the same effect as turning the terminal's power off and then back on except that the power-on self-test is not performed.

A hard reset does the following:

1. Rings the terminal's bell.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears all of alphanumeric memory.
5. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
6. Halts any datacomm transfers currently in progress, clears the datacomm buffer, and reinitializes the datacomm port according to the appropriate power-on datacomm configuration parameters.
7. Resets the terminal configuration menu parameters to values saved in non-volatile memory or to their default power-on values.
8. Resets certain operating modes and parameters as follows:
   - Disables display functions mode, caps mode, report mode, metric mode, data logging, and modify line.
   - Resets the left margin to column 1.
   - Resets the right margin to column 80.
   - Turns off the "insert character" function edit.
   - Resets the integral printer, if present.
   - Resets the User Keys to default values.
   - Resets record mode.
9. Resets certain operating modes in the HP 2623A:
   - Turns on alphanumeric display.
   - Turns on graphics display.
   - Clears graphics memory.
   - Turns off graphics cursor.
   - Enables graphics fucntions of the graphics/numeric keypad.

From the keyboard, you perform a hard reset by simultaneously pressing the SHIFT, CTRL and RESET keys.

From a program executing in a host computer, you perform a hard reset using the following escape sequence:

ESC E

BREAK

Pressing the BREAK key transmits a 200 ms space on the asynchronous data communications line, or sets the secondary channel low for 200 ms (depending on whether the terminal is in transmit or receive state). This serves as a "break" signal to interrupt computer operation.

BELL

The keyboard includes an embedded speaker for sounding an audible tone in response to the ASCII Bell (<BELL>) control code and for alerting the terminal operator when certain error conditions occur.

From the keyboard, you generate the Bell code by simultaneously pressing the CTRL and G keys.

From a program executing in a host computer, you trigger the bell tone by transmitting an ASCII Bell control code (decimal 7).

WAIT

From a user key or from a program executing in a host computer, you can cause the terminal to pause for approximately 1 second using the following escape sequence:

ESC [@

Multiple uses of this escape sequence in succession can be used to obtain virtually any desired time delay.

Note that while an ESC [@ is in effect, the cursor disappears from the screen, the keyboard is locked, and the passing of data from the datacomm firmware to display memory is inhibited.

For example, if you want to sound the bell tone twice in succession with a two-second delay between tones, you could do so using the following control sequence:

<BELL> ESC [@ ESC [@ <BELL>

MODEM DISCONNECT

You can direct the terminal to "hang up" the modem by sending an ESC [f. The terminal accomplishes the modem disconnect by lowering the TR/CD (Terminal Ready) line for 2 seconds.

Display Control

SECTION IV

INTRODUCTION

This section discusses the alphanumeric display control of the terminal. The graphics display control of the HP 2623A is discussed in Section X.

The display portion of the terminal consists of a display screen and display memory. The display cursor (a blinking underscore mark on the screen) indicates where the next character entered will appear. As you enter characters, each is displayed at the cursor position, the ASCII code for the character is recorded at the associated position in display memory, and the cursor moves to the next character position on the screen. As the screen becomes full, newly entered data causes existing lines to be forced off the screen. Data lines forced off the screen are still maintained in display memory and can subsequently be moved back onto the screen.

You can perform the following display control operations either locally from the keyboard or remotely from a program executing in a host computer:

- Move the cursor up, down, left, or right on the screen.
- Move the displayed data up or down in relation to the current cursor position. When a roll operation forces data off the top or bottom edge of the screen, additional data rolls onto the screen at the opposite edge from display memory.
- Change the content of the screen to the next or previous "page" of data in display memory. (A page consists of 24 lines.)
- Set or clear a left and right margin.
- Set or clear one or more tab stop positions.
- Move the cursor forward to the next tab stop position or backward to the preceding tab stop position.
- Enable or disable the inverse video, half bright, underline, and blinking, display enhancements.
- Change from one character set to another (Roman or Line Drawing, if present).

In addition, you can do the following screen edit operations either locally or remotely:

- Delete all characters from the current cursor position through the end of display memory.
- Delete the line containing the cursor (subsequent lines are rolled up).
- Change the line containing the cursor to all blanks.
- Turn off screen display except for softkey labels.
- Delete the character at the current cursor position.
- Insert a blank line immediately preceding (above) the line currently containing the cursor.
- Enable or disable "insert character" mode. When this editing mode is enabled, succeeding characters entered through the keyboard or received from the host computer are inserted to the left of the character at the current cursor position.

CURSOR CONTROL

The following topics describe how to alter the cursor/data relationship either manually by using the cursor control keys or programmatically by using escape sequences.

Home Up

Pressing the [key] key moves the cursor to the left margin in the top row of the screen and rolls the text in display memory down as far as possible so that the first line in display memory appears in the top row of the screen.

When format mode is enabled, the [key] key also rolls the text down as far as possible but leaves the cursor positioned at the beginning of the first unprotected field. If no fields have been defined, the cursor will appear at the first column of the first row on the screen.

To perform this function programmatically, use the following escape sequence:

ESC H

When memory lock is enabled, the [key] key rolls the text down as far as possible below the locked area of the screen, instead of below the top of the screen, and leaves the cursor positioned at the beginning of the first unlocked row on the screen. When both format mode and memory lock are active simultaneously, the cursor will go to the first unprotected field on the screen (including the locked area), after rolling all the text down.

NOTE

If memory lock is on and the cursor is within the locked area, [key] will cause the cursor to go down to the first character of the first line of text under the locked area, after rolling the text down.

Home Down

Pressing the Shift and ↖ keys moves the cursor to the left margin in the bottom line of the screen and rolls the text in display memory up as far as possible so that the last line in display memory appears immediately above the cursor position.

To perform this function programmatically, use the following escape sequence:

ESC F

Move Cursor Up

Each time you press the ↑ key, the cursor moves upward one row in the current column position. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the top row of the screen, pressing this key moves the cursor to the same column position in the bottom row of the screen.

To perform this function programmatically, use the following escape sequence:

ESC A

Move Cursor Down

Each time you press the ↓ key, the cursor moves downward one row in the current column position. If you hold the key down, the cursor movement continues row-by-row until the key is released. When the cursor is in the bottom row of the screen, pressing this key moves the cursor to the same column position in the top row of the screen.

To perform this function programmatically, use the following escape sequence:

ESC B

Move Cursor Right

Each time you press the → key, the cursor moves one column to the right in the current screen row. If you hold the key down, the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the rightmost column of the screen, pressing this key moves the cursor to the leftmost column in the next lower row (from the rightmost column in the bottom row of the screen, the cursor moves to the leftmost column in the top row of the screen).

To perform this function programmatically, use the following escape sequence:

ESC C

Move Cursor Left

Each time you press the ← key, the cursor moves one column to the left in the current screen row. If you hold the key down, the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the leftmost column of the screen, pressing this key moves the cursor to the rightmost column in the next higher row (from the leftmost column in the top row of the screen, the cursor moves to the rightmost column in the bottom row of the screen).

To perform this function programmatically, use the following escape sequence:

ESC D

Roll Text Up

Each time you press the Roll Up key, the text in display memory rolls up one row on the screen. The top row rolls off the screen, the remaining data rolls up one line on the screen, and a new line of data rolls from display memory into the bottom line of the screen. If you hold this key down, the text continues to roll upward until you release the key or until the final line of data in display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll up" function is illustrated in figure 4-1A.

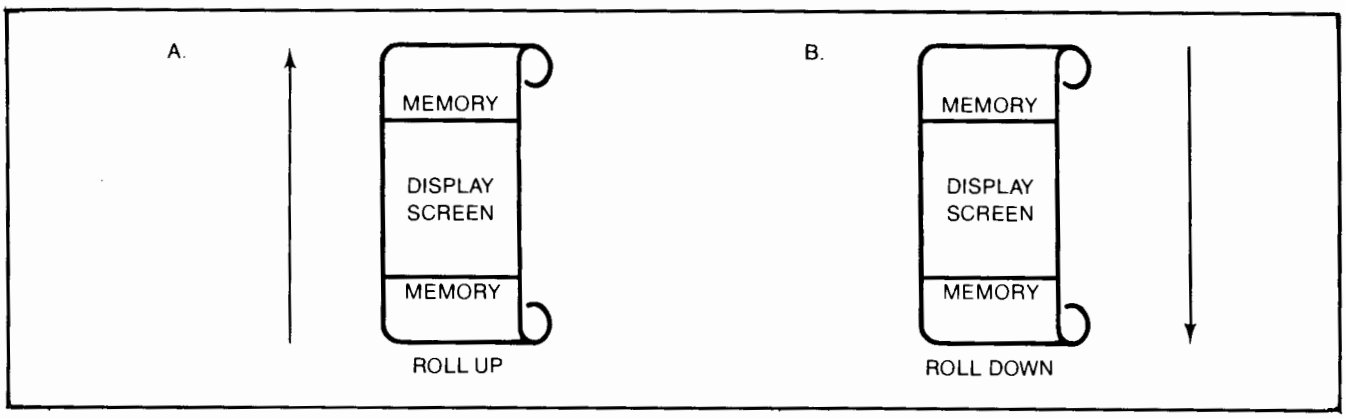


Figure 4-1. The "Roll" Data Functions

In the configuration and user softkey definition menus, this key is disabled. In memory lock mode, the unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the screen.

To perform this function programmatically, use the following escape sequence:

ESC&S

Roll Text Down

Each time you press the ROLL DOWN key, the text in display memory rolls down one row on the screen. The bottom row rolls off the screen, the remaining data rolls down one line on the screen, and a new line of data rolls from display memory into the top line of the screen. If you hold this key down, the text continues to roll downward until you release the key or until the first line of data in display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll down" function is illustrated in figure 4-1B.

In the configuration and user softkey definition menus, this key is disabled. In memory lock mode, the unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the screen. The cursor position relative to the screen remains unchanged after this operation.

To perform this function programmatically, use the following escape sequence:

ESC&T

Next Page/Previous Page

The data in display memory can be accessed (displayed on the screen) in blocks that are known as "pages". A page consists of 24 lines of data. The current page is that sequence of lines which appears on the screen at any given time. The previous page is the preceding 24 lines in display memory. The next page is the succeeding 24 lines in display memory.

The concept of display "pages" is illustrated in figure 4-2.

Pressing the NEXT PAGE key rolls the text in display memory up so that the next page of data replaces the current page on the screen. If you hold the key down, the operation is repeated so the final line in display memory appears in the top line of the screen.

In the configuration and user softkey definition menus, this key is disabled.

To perform the "next page" function programmatically, use the following escape sequence:

ESC&U

Pressing the PREV PAGE key rolls the text in display memory down so that the previous page of data replaces the current page on the screen. If you hold the key down, the operation is repeated so the first line in display memory appears in the top line of the screen.

In the configuration and user softkey definition menus, this key is disabled.

To perform the "previous page" function programmatically, use the following escape sequence:

ESC&V

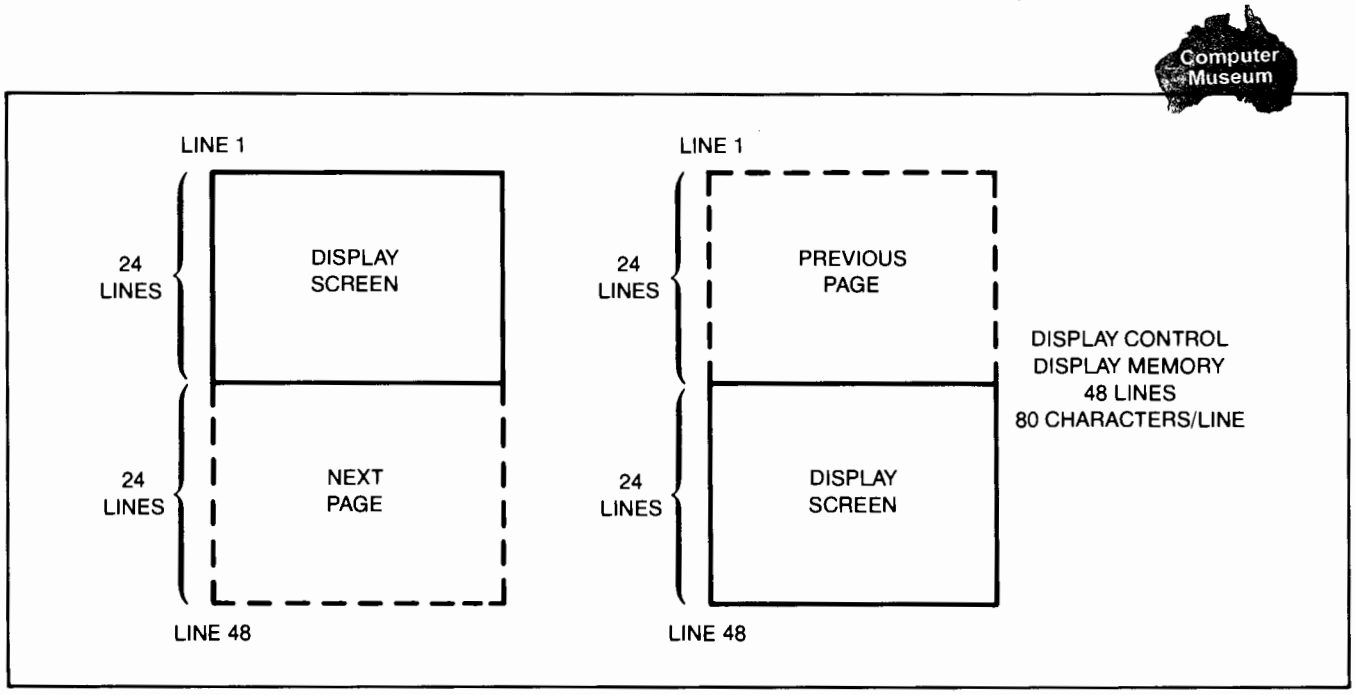


Figure 4-2. Previous Page and Next Page Concepts

At the completion of the "next page" or "previous page" function, the cursor is positioned at the left margin in the top line of the screen.

If format mode is on, the cursor will go to the first unprotected field on the new page.

MEMORY ADDRESSING SCHEME

Display memory positions can be addressed using absolute or relative coordinate values. Display memory is made up of 80 columns (0–79) and 48 lines (0–47). There can be as many as 48 lines of 80 characters (2 screens). The amount of memory in the terminal can be determined from byte 0 of the primary terminal status (refer to Section VII). The types of addressing available are:

- Absolute
- Screen Relative
- Cursor Relative

ROW ADDRESSING. Figure 4-3 illustrates the way the three types of addressing affect row or line numbers. The cursor is shown positioned in the fourth row on the screen. Screen row 0 is currently at row 6 of display memory. In order to reposition the cursor to the first line of the screen the following three destination rows could be used:

a. Absolute: row 6
b. Screen Relative: row 0
c. Cursor Relative: row −3

COLUMN ADDRESSING. Column addressing is accomplished in a manner similar to row addressing. There is no difference between screen and cursor relative column addressing. Figure 4-4 illustrates the difference between absolute and relative addressing. The cursor is shown in column 5.

Whenever the row or column addresses exceed those available, the largest possible value is substituted. In screen relative addressing, the cursor cannot be moved to a row position that is not currently displayed. For example, in figure 4-3c a relative row address of −10 would cause the cursor to be positioned at the top of the current screen (relative row −3). Column positions are limited to the available screen positions (0 to 79 in figure 4-4a and −5 to 74 in figure 4-4b). The cursor cannot be wrapped around from column 0 to column 79 by specifying large negative values for relative column positions.

Cursor Sensing

The current position of the screen cursor can be sensed. The position returned can be the absolute position in display memory or the location relative to the current screen position. (Absolute and relative addresses are discussed under Cursor Addressing.)

Absolute Sensing ESC t a

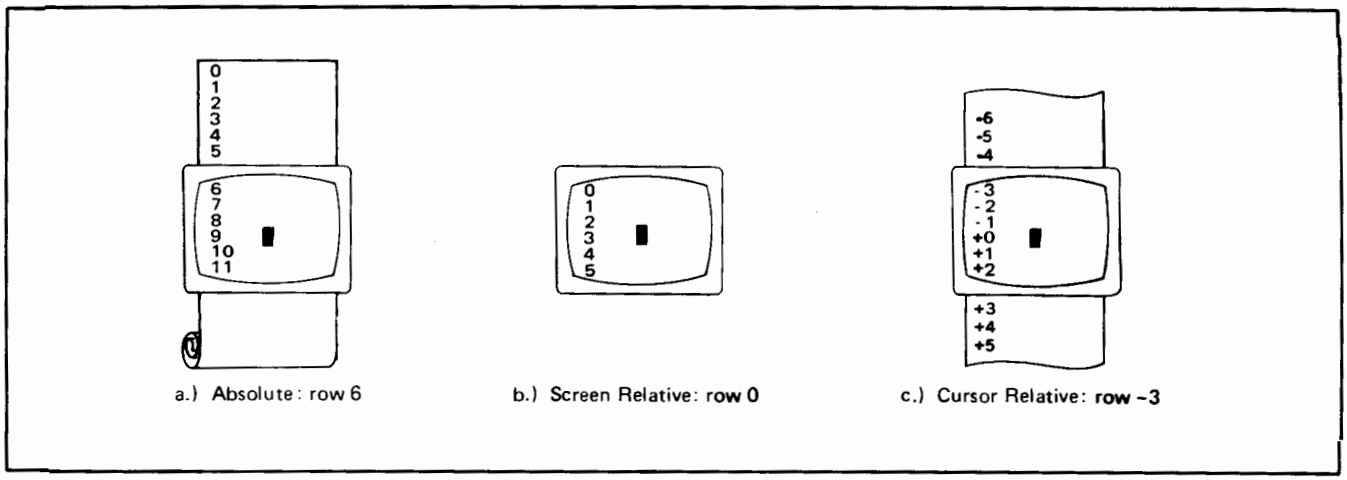
Example: The cursor is at column 20, row 40.
computer: ESC t a
terminal: ESC t & a 020c 040R

Relative Sensing ESC t '

Example: The cursor is again at column 20, row 40, but screen row 0 begins at row 35 of display memory.
computer: ESC t '
terminal: ESC t & a 020c 005Y

Cursor Positioning

The cursor can be positioned directly by giving memory or screen coordinates, or by sending the escape codes for any of the keyboard cursor positioning operations.



a.) Absolute: row 6 b.) Screen Relative: row 0 c.) Cursor Relative: row -3

Figure 4-3. Row Addressing

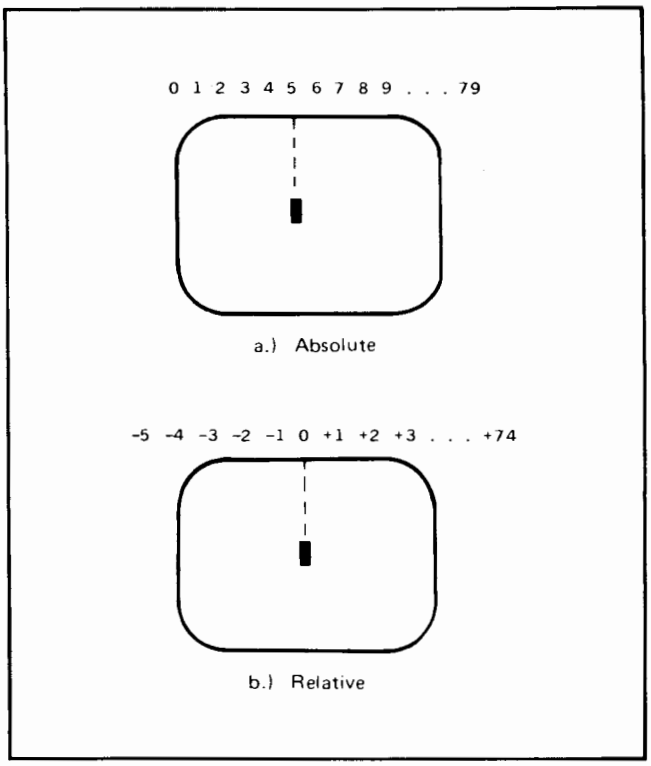


Figure 4-4. Column Addressing

Screen Relative Addressing

To move the cursor to any character position on the screen, use any of the following escape sequences:

```
Ec&a <column number> c <row number> Y
Ec&a <row number> y <column number> C
Ec&a <column number> C
Ec&a <row number> Y
```

where:

`<column number>`	is a decimal number specifying the screen column to which you wish to move the cursor. Zero specifies the leftmost column.
`<row number>`	is a decimal number specifying the screen row (0–23) to which you wish to move the cursor. Zero specifies the top row of the screen; 23 specifies the bottom row.

When using the above escape sequences, the data on the screen always remains unchanged.

If you specify only a `<column number>`, the cursor remains in the current row. Similarly, if you specify only a `<row number>`, the cursor remains in the current column.

Example: The following escape sequence moves the cursor to the 20th column of the 7th row on the screen:

```
Ec&a 6y 19C
```

Absolute Addressing

You can specify the location of any character within display memory by supplying absolute row and column coordinates. (Note that this function is disabled when memory lock mode is on.) To move the cursor to another character position using absolute addressing, use any of the following escape sequences:

```
Ec&a <column number> c <row number> R
Ec&a <row number> r <column number> C
Ec&a <column number> C
Ec&a <row number> R
```

where:

`<column number>`	is a decimal number (0–79) specifying the column coordinate (within display memory) of the character at which you want the cursor positioned. Zero specifies the first (leftmost) column in display memory, 79 the rightmost column.
`<row number>`	is a decimal number (0–47) specifying the row coordinate (within display memory) of the character at which you want the cursor positioned. Zero specifies the first (top) row in display memory, 47 specifies the last.

When using the above escape sequences, the data visible on the screen will (if necessary) be rolled up or down in order to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the absolute row coordinate is less than that of the top line currently visible on the screen, the cursor moves to the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen.
- If the absolute row coordinate exceeds that of the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen.

If you specify only a `<column number>`, the cursor remains in the current row. Similarly, if you specify only a `<row number>`, the cursor remains in the current column.

Example: The following escape sequence moves the cursor (and rolls the text if neccessary) so that it is positioned at the character residing in the 60th column of the 27th row in display memory:

```
Ec&a 26r 59C
```

Cursor Relative Addressing

You can specify the location of any character within display memory by supplying row and column coordinates that are relative to the current cursor position. (Note that this function is disabled when memory lock mode is on.) To move the cursor to another character position using cursor relative addressing, use any of the following escape sequences:

```
Esc&a +/- <column number> c +/- <row number> R
Esc&a +/- <row number> r +/- <column number> C
Esc&a +/- <column number> C
Esc&a +/- <row number> R
```

where:

`<column number>`	is a decimal number specifying the relative column to which you wish to move the cursor. A positive number specifies how many columns to the right you wish to move the cursor; a negative number specifies how many columns to the left.
`<row number>`	is a decimal number specifying the relative row to which you wish to move the cursor. A positive number specifies how many rows down you wish to move the cursor; a negative number specifies how many rows upward.

When using the preceding escape sequences, the data will (if necessary) be rolled up or down in order to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified cursor relative row precedes the top line currently visible on the screen, the cursor moves to the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen.
- If the specified cursor relative row follows the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen.

If you specify only a `<column number>` the cursor remains in the current row. Similarly, if you specify only a `<row number>` the cursor remains in the current column.

Example: The following escape sequence moves the cursor (and rolls the text if necessary) so that it is positioned at the character residing 15 columns to the right and 25 rows above the current cursor position within display memory:

```
Esc&a +15c -25R
```

Combining Absolute And Relative Addressing

You may use a combination of screen relative, absolute and cursor relative addressing within a single escape sequence.

Example: Move the cursor (and roll the text if necessary) so that it is positioned at the character residing in the 70th column of the 18th row below the current cursor position.

```
Esc&a 69c +18R
```

Example: Move the cursor so that it is positioned at the character residing 15 columns to the left of the current cursor position in the 4th row currently visible on the screen.

```
Esc&a -15c 3Y
```

Example: Move the cursor (and roll the text up or down if necessary) so that it is positioned at the character residing in the 10th column of absolute row 48 in display memory.

```
Esc&a 9c 47R
```

EDIT OPERATIONS

You can edit data displayed on the screen by simply overstriking the old data. In addition, the terminal provides the following edit functions which can be enabled and disabled either manually by using the edit control keys or programmatically by using escape sequences:

- Insert Line.
- Delete Line.
- Insert Character.
- Delete Character.
- Clear Display.
- Clear Line.

Insert Line

When you use the insert line edit function, the text line containing the cursor and all text lines below it roll downward one line, a blank line is inserted in the screen row containing the cursor, and the cursor moves to the left margin of the blank line. Note that when memory lock mode is active, inserting a line within the locked area of the screen does not change the size of the locked area.

From the keyboard, each time you press the INSERT LINE key the terminal inserts one blank line. If you hold the key down, the terminal continues to insert blank lines until the key is released.

This edit function is disabled in format mode, and is disabled in the configuration and user softkeys definition menus.

NOTE

When display memory (48 rows) is full, inserting a line will cause data to be lost. The first line in display memory will always be the one to be released unless it happens to be on the screen, in which case the last line in display memory will be released.

From a program executing in a host computer, you insert a blank line at the current cursor position using the following escape sequence:

ESC L

Delete Line

When you use the delete line edit function, the text line containing the cursor is deleted from display memory, all text lines below it roll upward one row, and the cursor moves to the left margin. Note that when memory lock mode is active, deleting a line within the locked area does not change the size of the locked area.

From the keyboard, each time you press the DELETE LINE key the terminal deletes one line of text. If you hold the key down, the terminal continues to delete text lines until the key is released or until there are no subsequent text lines remaining in display memory. In the latter case, pressing or continuing to hold down this key has no further effect.

This edit function is disabled in format mode, and is disabled in the configuration and user softkeys definition menus.

From a program executing in a host computer, you delete the text line at the current cursor position using the following escape sequence:

ESC M

Insert Character

When the insert character editing function is enabled, characters entered through the keyboard or received from the host computer are inserted into display memory at the cursor position. Each time a character is inserted, the cursor and all characters from the current cursor position through the right margin move one column to the right. Characters that are forced over the right margin are lost. When the cursor reaches the right margin, it moves to the left margin in the next lower line and the insert character function continues from that point.

This edit function is meant to be used within that portion of the screen delineated by the left and right margins. If you position the cursor to the left of the left margin, the insert character function works as described above. If you position the cursor beyond the right margin, however, the insert character function affects those characters between the current cursor position and the right boundary of the screen. In such a case, when the cursor reaches the right boundary of the screen, it moves to the left margin in the next lower line and the insert character function continues from that point as described in the first paragraph above.

The movement of existing characters during an "insert character" editing operation is illustrated in figure 4-5.

When format mode is off, any unprotected, alternate character set, and/or video enhancement fields to the right of the cursor move to the right with the displayable characters. If the cursor is positioned within any such field the insert character function extends the range of the field by one position for each character inserted. Block terminators at or to the right of the cursor position move to the right along with the displayable characters.

When format mode is on and the cursor is positioned within an unprotected field, the insert character function affects only those characters from the cursor position through the end of the current subfield. Block terminators are treated the same as when format mode is off. If the cursor is not within an unprotected field, it automatically moves to the first character position of the next subsequent unprotected field when the first character is inserted.

In the user softkeys definition menu, insert character acts the same as in format mode; insert character is disabled in a configuration menu.

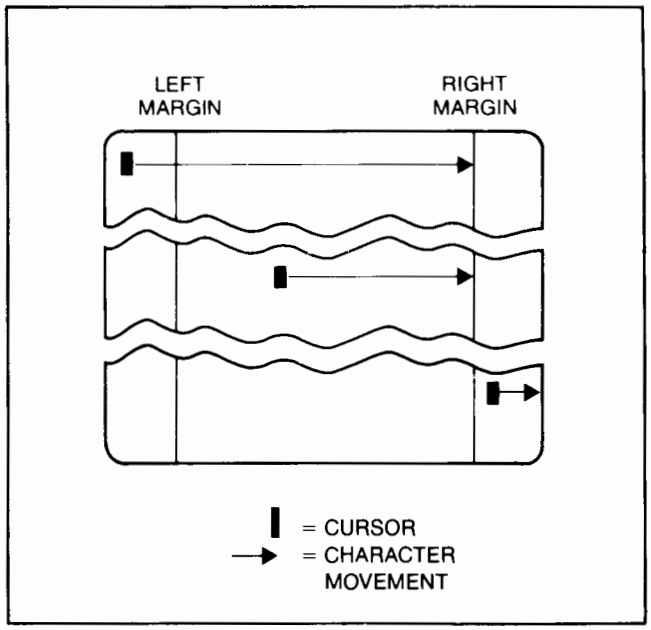


Figure 4-5. Character Insert with Margins

From the keyboard, you enable and disable the insert character editing function using the INSERT CHAR key. When enabled, the characters "IC" are displayed in the status line at the bottom of the screen.

From a program executing in a host computer, you enable and disable the insert character editing function using the following escape sequences:

ENABLE: ESC Q
DISABLE: ESC R

Delete Character

When you use the delete character edit function, the cursor remains stationary, the character at the cursor position is deleted, all characters between the cursor and the right margin move left one column, and a blank moves into the line from the right margin.

This edit function is meant to be used within that portion of the screen delineated by the left and right margins. If you position the cursor to the left of the left margin, the delete character function works as described above. If you position the cursor beyond the right margin, however, the delete character function affects those characters from the current cursor position through the right boundary of the screen.

The movement of existing characters during a "delete character" editing operation is illustrated in figure 4-6.

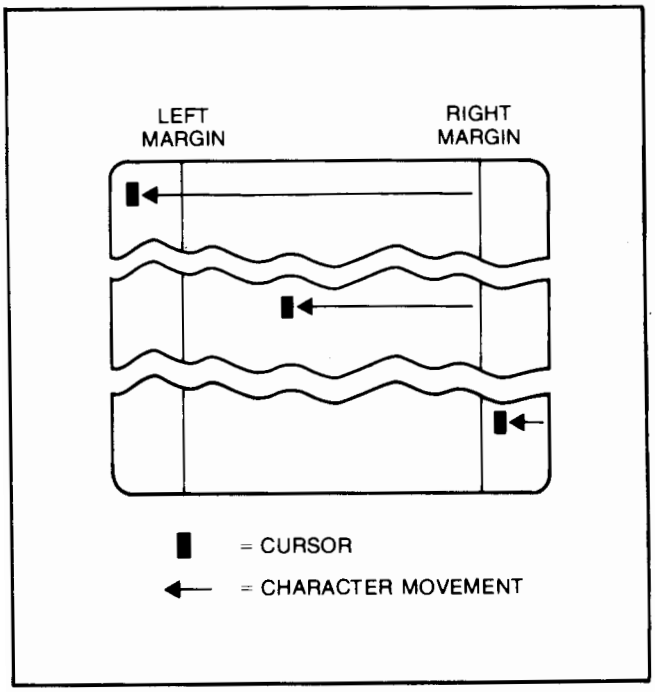


Figure 4-6. Character Delete with Margins

When format mode is off, any unprotected, alternate character set, and/or video enhancement fields to the right of the cursor move to the left with the displayable characters. If the cursor is positioned within any such field, the delete character function shortens the range of the field by one position for each character deleted. Deleting the first character position of an unprotected field changes the rest of the field to protected. Deleting characters at the start of, or within, a video enhancement and/or alternate character set field does NOT alter the characteristics of the rest of the field. Block terminators to the right of the cursor move to the left along with the displayable characters and are deleted if they are at the cursor position when this function is executed.

When format mode is on and the cursor is positioned within an unprotected field, this function affects only those characters from the cursor position through the end of the current subfield. If the subfield definition also includes a video enhancement and/or an alternate character set, those characteristics are NOT altered by the delete character function. Block terminators are treated the same as when format mode is off. If the cursor is not within a protected field, the delete character function has no effect.

In the user softkeys definition menu, delete character acts the same as in format mode; delete character is disabled in a configuration menu.

From the keyboard, each time you press the DELETE CHAR key the terminal deletes one character. If you hold the key down, the terminal continues to delete characters until either the key is released or there are no non-blank characters between the cursor position and the right margin. In the latter case, pressing or continuing to hold down this key has no further effect.

From a program executing in a host computer, you delete the character at the current cursor position using the following escape sequence:

ESC P

Clear Display

When format mode is off, pressing the CLEAR DISPLY key deletes all displaying and non-displaying characters from the current cursor position through the end of display memory.

When format mode is on, pressing the CLEAR DISPLY key deletes all unprotected displaying and non-displaying characters, all unprotected video enhancements, and any unprotected line drawing characters from the current cursor position through the end of display memory.

This key is disabled in the user softkeys definition and configuration menus.

To perform this function programmatically, use the following escape sequence:

ESC J

Clear Line

When format mode is off, pressing the CLEAR LINE key deletes all displaying and non-displaying characters from the current cursor position through the end of the current line.

When format mode is on and the cursor is positioned within an unprotected field, pressing the CLEAR LINE key deletes all displaying and non-displaying characters and all unprotected video enhancements from the current cursor position through the end of the current field. If the cursor is not within an unprotected field, the CLEAR LINE key has no effect.

In the user softkeys definition menu, clear line acts the same as in format mode; clear line is disabled in a configuration menu.

To perform this function programmatically, use the following escape sequence:

ESC K

SETTING AND CLEARING MARGINS

You can redefine the left and/or right margin. These margins affect the cursor positioning for certain functions (such as carriage return, home up, home down, etc.) and establish operational bounds for the insert character and delete character functions. In addition, the left margin is always an implicit tab stop. Data to the left of the left margin or to the right of the right margin is still accessible. Data transfers from display memory to a host computer or to a printer are performed without regard to margins. Format mode, when enabled, clears the margins, creating an 80 character line (1–80).

When you are entering data through the keyboard and the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (note that this operating characteristic can be disabled through the use of the "InhEolWrp" terminal configuration parameter; refer to Section II). When you press RETURN, the cursor moves to the left margin in the current line if auto line feed mode is disabled or to the left margin in the next lower line if auto line feed mode is enabled.

When data is being received from a host computer, it enters display memory only within the defined margins. When the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (as mentioned above, this operating characteristic can be disabled through the use of the "InhEolWrp" configuration parameter). When an ASCII CR control code is received, the cursor always moves to the left margin in the current line regardless of whether or not auto line feed mode is enabled.

From the keyboard, you set and clear the margins using the **margins/tab/col** set of function keys. To get to that set, use the following keystroke sequence:

[AIDS] [f2] margins/ tab/col

This changes the function key labels to the following:

[f1]	[f2]	[f3]	[f4]
START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS
[f5]	[f6]	[f7]	[f8]
LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	

To set the left or right margin, move the cursor to the desired column and then press the appropriate function key (f5 or f6). To reset the left margin to column 1 and the right margin to column 80, press f7 .

If you attempt to set either margin incorrectly with relation to the other (e.g., the right margin to the left of the left margin), the terminal rejects it with an audible "beep".

From a program executing in a host computer, you set and clear the margins using the following escape sequences:

SET LEFT MARGIN: ESC 4
SET RIGHT MARGIN: ESC 5
CLEAR ALL MARGINS: ESC 9

The first two escape sequences set the left and right margin (respectively) at the current cursor position. Before using them, therefore, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

SETTING AND CLEARING TABS

You can define a series of tab stops to which you can move the cursor using the tab and back tab functions (described as separate topics later in this section).

From the keyboard, you set and clear tab stops using the **margins/tab/col** set of function keys. To get to that set, use the following keystroke sequence:

[AIDS] [f2] margins/ tab/col

This changes the function key labels to the following:

[f1]	[f2]	[f3]	[f4]
START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS
[f5]	[f6]	[f7]	[f8]
LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	

To set a tab stop, move the cursor to the desired column and then press f2 . To clear a tab stop, move the cursor to the particular tab stop position and then press f2 . To clear all existing tab stops, press f4 . Note that the left margin is always an implicit tab stop and is not affected by f4 .

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored when the tab or back tab functions are performed.

From a program executing in a host computer, you set and clear tab stops using the following escape sequences:

SET TAB: ESC 1
CLEAR TAB: ESC 2
CLEAR ALL TABS: ESC 3

The first two escape sequences set and clear (respectively) a tab stop at the current cursor position. Before using them, therefore, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

TAB

From the keyboard, you can move the cursor ahead to the next subsequent tab stop using the TAB key (or the TAB key in the numeric pad). In format mode, tab moves the cursor to the beginning of the next unprotected field. The last field wraps around the beginning of the first field. Tab acts similarly in the user softkeys definition menu and the configuration menu.

From a program executing in a host computer, you can move the cursor ahead to the next tab stop issuing either an ASCII HT control code (decimal 9; Control "I") or the following escape sequence:

ESC I

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored by the tab function.

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the right of the rightmost tab stop, the tab function moves the cursor to the left margin in the next lower line. When the cursor is positioned to the left of the left margin, however, the tab function advances the cursor to the first explicit tab stop in the line (or to the left margin in the next lower line if no explicit tab stops are defined). Note that tabbing the cursor to the next line is the equivalent of a linefeed.

BACK TAB

From the keyboard you can move the cursor backward to the previous tab stop using the SHIFT and TAB keys (or the TAB key in the numeric pad).

In format mode, configuration menus, and user keys definition menu, the cursor, if within a field, will move to the beginning of the field; otherwise it will move to the first character of the previous unprotected field.

From a program executing in a host computer you can move the cursor backward to the previous tab stop using the following escape sequence:

ESC i

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored by the back tab function.

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the left of the left margin, the back tab function moves the cursor to the rightmost tab stop in the next higher line.

Performing a back tab with the cursor on the left margin of the first row on the screen (or the first unlocked row if memory lock mode is active) is equivalent to performing a roll down.

SCREEN BLANKING

From a program executing in a host computer you can turn off and on the alphanumeric display video, excluding the softkey labels. This feature may be used to turn off the display video while a form is being drawn into alphanumeric memory. After the form is completed, the video may be turned on. The escape sequences follow:

Turn off display video: ESC &w13F
Turn on display video: ESC &w12F

DISPLAY ENHANCEMENTS

The terminal includes as a standard feature the following display enhancement capabilities:

- Inverse Video—black characters are displayed against a white background.
- Underline Video—characters are underscored.
- Blink Video—characters blink on and off.
- Half Bright—characters (or background for inverse video) are displayed at half intensity.

You use these enhancements on a field basis. They may be used separately or in any combination. When used, they cause control bits to be set within display memory. If the content of display memory is subsequently transmitted in block mode to a host computer, these control bits are translated into escape sequences which are transmitted along with the displayable text characters.

From a program executing in a host computer, or from the keyboard, you enable and disable the various video enhancements by embedding escape sequences within the data. The general form of the escape sequence is as follows:

ESC &d <enhancement code>

where enhancement code is one of the uppercase letters A through O specifying the desired enhancement(s) or an @ to specify end of enhancement as follows:

Enhancement Character

	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright									x	x	x	x	x	x	x	x
Underline					x	x	x	x					x	x	x	x
Inverse Video			x	x			x	x			x	x			x	x
Blinking		x		x		x		x		x		x		x		x
End Enhancement	x															

Note that the escape sequence for "end enhancement" (ESC&d @) or the escape sequence for another video enhancement, will end the previous enhancement.

Example: Define columns 10 through 14 of line 5 to be inverse video and blinking.

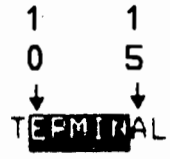
Step 1. Position the cursor at column 10 in line 5.

Step 2. Enter ESC&d C.

Step 3. Move the cursor to column 15 in line 5.

Step 4. Enter ESC&d @ (this ends the enhancements). The field should be white.

Step 5. Enter the word TERMINAL beginning in column 9 of line 5. It should appear as shown below. The characters "ERMIN" should be in inverse video and blinking.



You may want to enter some frequently used enhancements into the user keys for ease in entering the enhancements onto the display.

Example: Enter Underline, Half-bright Inverse, Inverse, and End Enhancement escape sequences into the f1 through f4 user keys (figure 4-7).

Step 1. Press Shift USER KEYS to display the menu.

Step 2. Press "NEXT CHOICE" f2 until "L" is displayed in the attribute field of f1 for local operation.

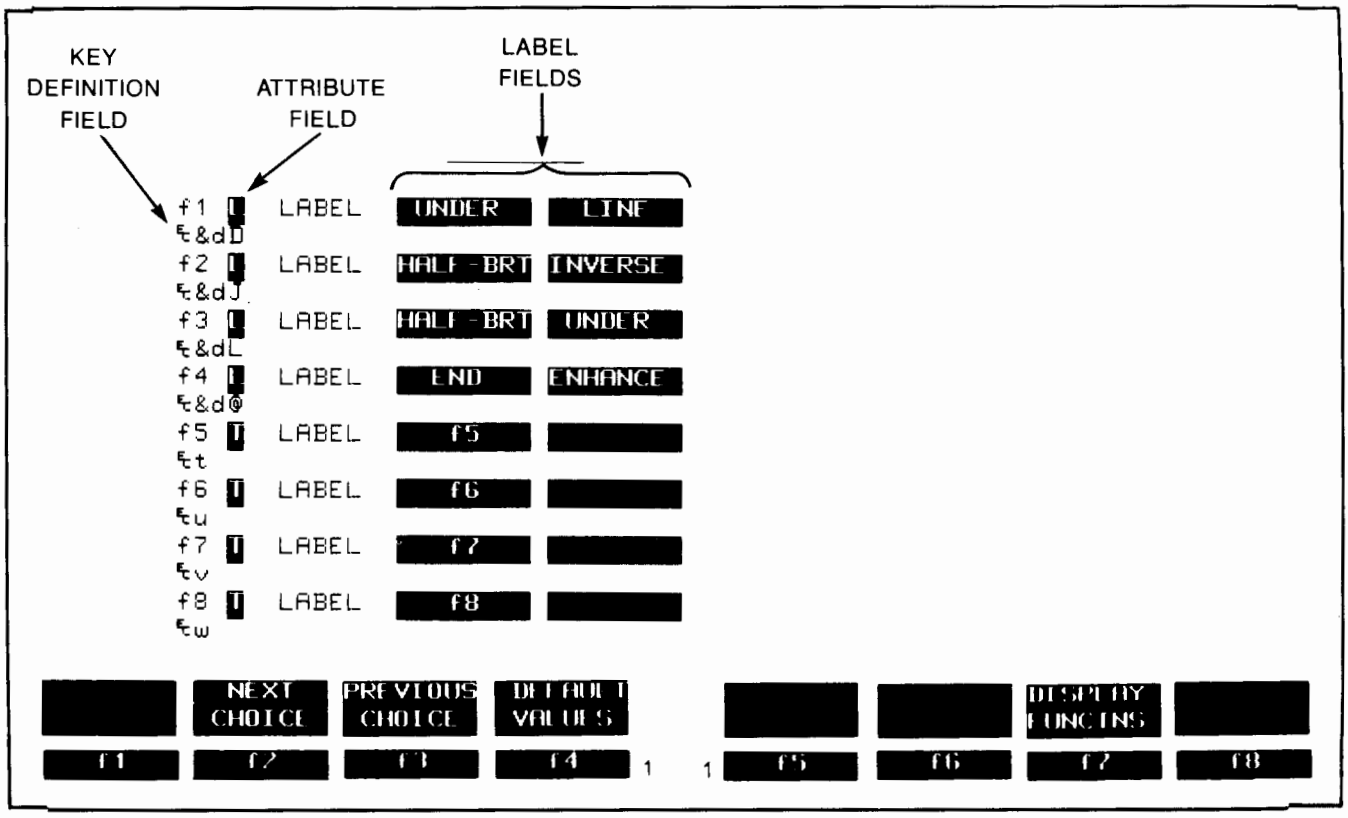


Figure 4-7. Menu After Entering Four Enhancement Escape Sequences

Step 3. Press TAB to position the cursor to the label field; then, type a meaningful label to represent the enhancement (e.g., "UNDER" in the first label and "LINE" in the second label field).

Step 4. Press TAB to position the cursor to the definition field; then, press the "DISPLAY FUNCTIONS" key. Now, enter the escape sequence for the enhancement; then, turn off display functions. (Display functions is turned on to enter the "&", escape, character.)

Step 5. Tab to the next key fields, and enter the appropriate data in a manner described above.

Step 6. When you have finished entering the data into the menu, press USER KEYS to return to the normal display with the user key labels.

Step 7. You many now turn on any enhancement specified in the user keys by positioning the cursor to where you want the enhancement to begin; then, press the appropriate user key. To turn off the enhancement, press the user key containing the end enhancement escape sequence.

DESIGNING AND USING FORMS

With the terminal, you can design elaborate data entry forms constructed of varying line types from the optional line drawing set and containing alphanumeric annotations and protected and unprotected fields.

When format mode is enabled, the cursor automatically moves to the start of the first unprotected field in the form. Henceforth, the terminal operator can only enter data into those portions of the display screen which lie within unprotected fields, the remainder of the screen is protected. When the operator enters a character into the last position of a field, the cursor advances to the start of the next unprotected field. In addition, the TAB and SHIFT TAB keys can be used to move the cursor to the start of the preceding or next unprotected field. If the cursor is within a protected field, it automatically advances to the start of the next unprotected field when the operator attempts to type a data character.

You enable and disable format mode programmatically by using the following escape sequences:

ENABLE: &W
DISABLE: &X

These sequences may be entered through the keyboard, executed from within a user key definition, or issued from a program in a host computer.

There are three major steps to creating data entry forms:

1. Create the linear structure of the form on the screen using the optional line drawing set.
2. Define the various unprotected fields within the form.
3. Programmatically read the completed form and store it in the host computer for future use.

DRAWING FORMS

In order to draw lines with your terminal, you need option 202 of the HP 2622A or HP 2623A terminal. Option 202 is equipped with two character sets: the standard USASCII character set and the Line Drawing character set.

With the terminal configured to its default state (i.e., set for 7-bit ASCII–Ascii 8-bit is NO in the terminal configuration menu), the standard USASCII set is defined as the Base set, and the Line Drawing set is defined as the alternate character set. The elements of the Line Drawing set and their relationship to the terminal's standard USASCII keyboard are illustrated in figure 4-7.

Note that the relationship of some line drawing set elements to the keys varies if the terminal is configured to another national character set. (Refer to the Terminal Configuration Menu discussed in Section II for the terminal's present configuration.) Appendix B provides a description of the national keyboard options and a code chart which cross-references the keyboard character to the line drawing set character.

The first step in generating a data entry form is to create the linear structure of the form on the screen along with any constant alphanumeric annotations such as the form's title and the row/column headings. You do this using the Line Drawing and standard USASCII character sets.

When you are designing a form through the keyboard, you use ASCII ^N (control-N) and ^O (control-O) codes to switch back and forth between the Line Drawing and USASCII character sets.

You switch from the Base set (USASCII) to the Line Drawing set by issuing an ASCII ^N code (control-N) and you switch from the Line Drawing set back to the Base set by issuing an ASCII ^O code (control-O). Note that the ^N code affects only those characters from the current cursor position through the next ^N or ^O code or through the end of the line, whichever occurs first. Consequently, if the Line Drawing set is enabled at the end of one line on the screen and you also want it enabled at the start of the next line, you will have to explicitly issue another ^N code at the start of the second line.

Figure 4-8 illustrates a sample form and identifies the keystrokes used for generating the various different types of line segments. Figure 4-9 shows the same form as it actually appears on the terminal's screen. Figure 4-10 shows the Base set equivalent characters for the entire form structure.

One approach to generating a form structure through the keyboard is to load two of the user keys with the ^N and ^O codes (control-N and control-O, respectively, with display functions mode enabled), define both as Local keys, define

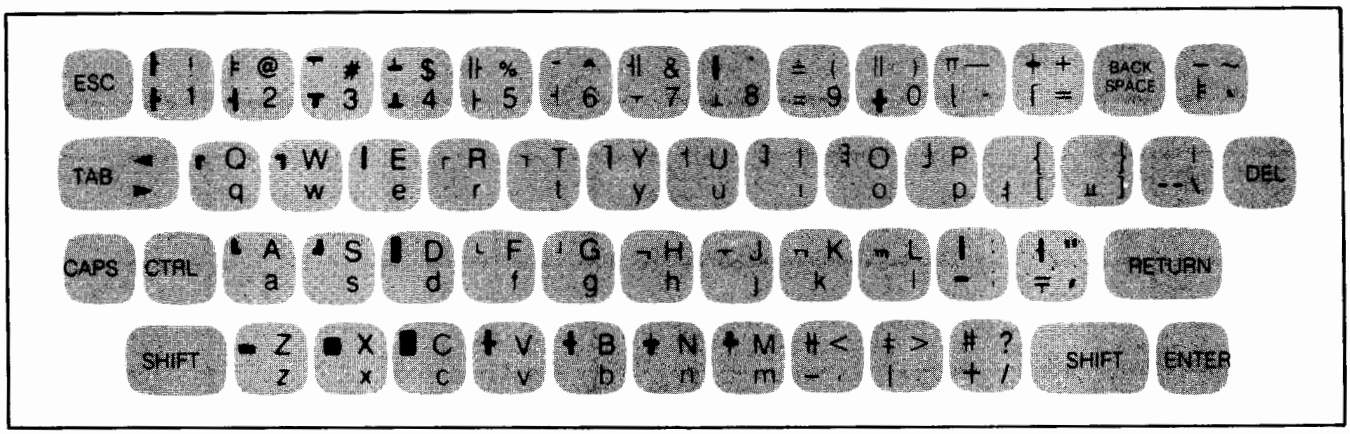


Figure 4-7. Line Drawing Set Elements

their Labels as "`Line Draw`" and "`Base Set`", respectively, and enable them by pressing USER KEYS. Then draw the form structure and alphanumeric annotations using the Base set as illustrated in figure 4-10. As the form is evolving, use the cursor control keys and the two user keys to switch the linear structure portions of the form to the Line Drawing character set. When doing this, however, be sure that those portions of the form that will be used for data fields are set to the Base set (figure 4-11 shows three lines of the sample form and the various points at which you would use the $^{5}_{6}$ and $^{5}_{1}$ codes). You may also, if you wish, load some of the more repetitive line definitions (such as the second and third lines in figure 4-11) into user keys to speed up the drawing of the main body of the form.

Video enhancements can be incorporated into the form using the user keys. Define the keys as "local", the labels as "`half-brt/inverse`", "`full-brt/inverse`", "`under/line`", "`enhance/off`", etc. The escape sequences are given previously in this section under "DISPLAY ENHANCEMENTS". To begin an enhancement, position the cursor on the screen; then, press the user key containing the desired enhancement. Next, to end the enhancement, position the cursor to where you want the enhancement to end (must be in the same line); then, press the key containing the escape sequence for ending the enhancement.

A program running in the host computer can also be used to draw forms on the terminal display screen.

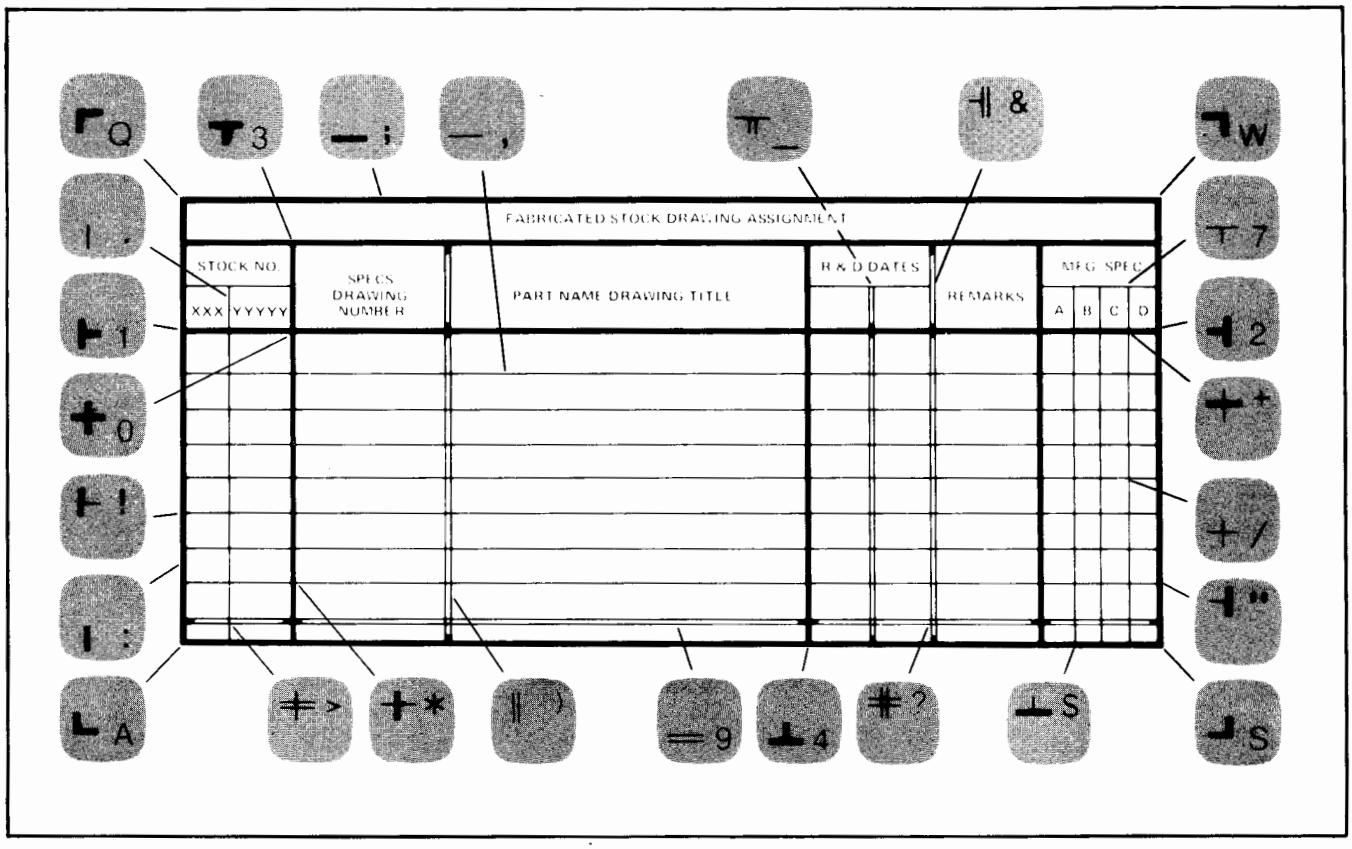
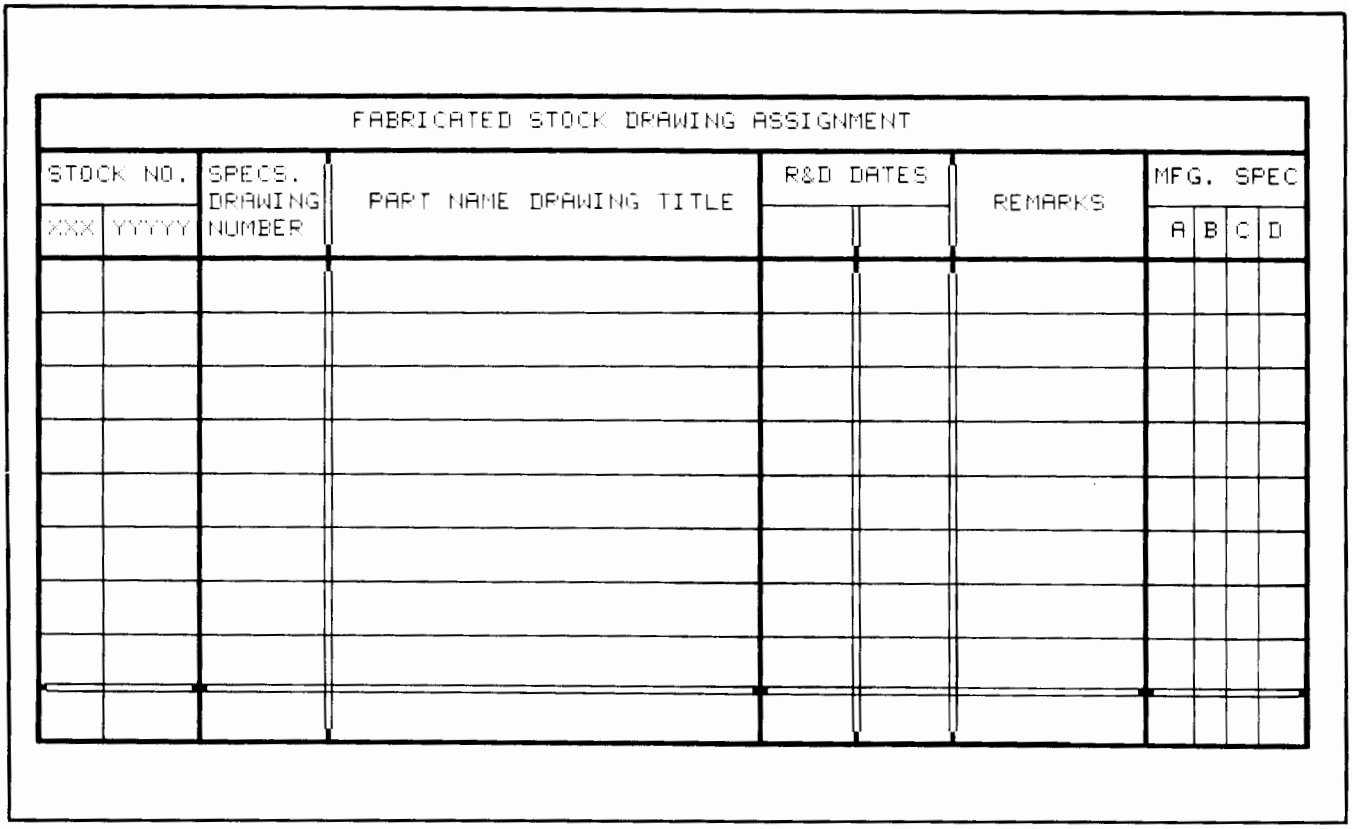


Figure 4-8. Sample Data Entry Form



FABRICATED STOCK DRAWING ASSIGNMENT

STOCK NO.		SPECS. DRAWING NUMBER	PART NAME DRAWING TITLE	R&D DATES		REMARKS	MFG. SPEC			
XXX	YYYYY						A	B	C	D

Figure 4-9. Completed Data Entry Form

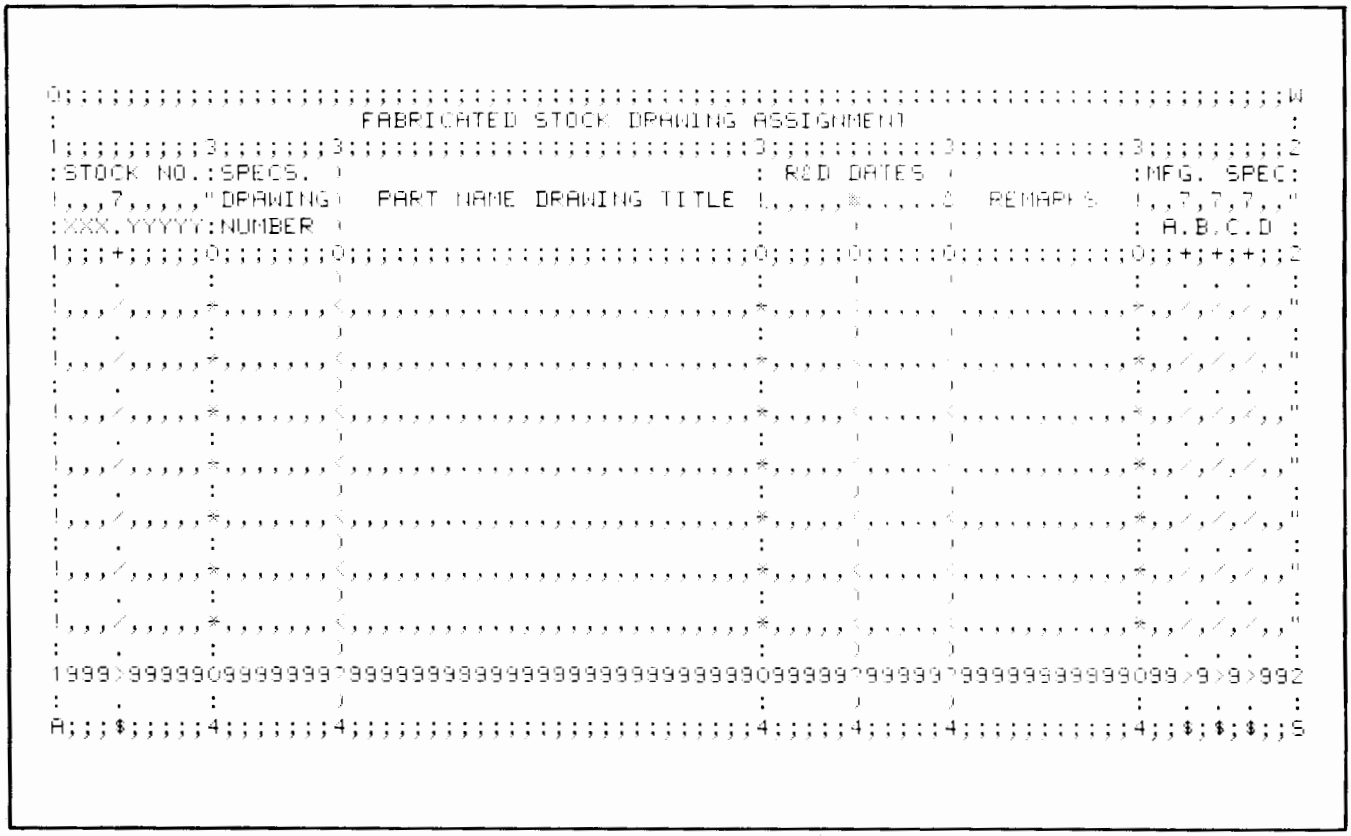


Figure 4-10. Base Set Equivalents for Data Entry Form Illustrated in Figure 4-9.

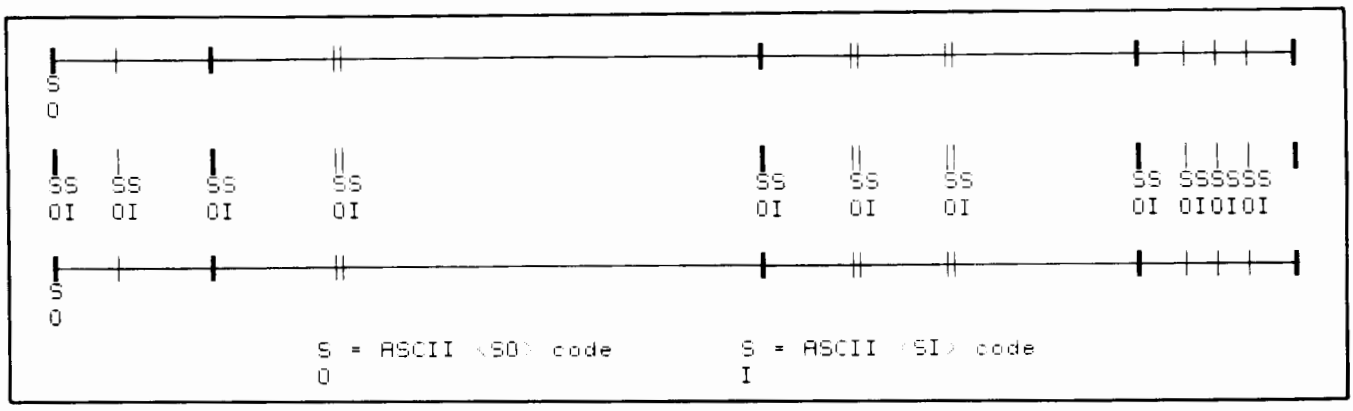


Figure 4-11. Use of Shift-In and Shift-Out Codes

The program must first define the line drawing character set as the alternate character set by sending an ESC)B to the terminal.

The program then must go through the same process of shifting out of the base character set (^N = control-N = decimal 14) to draw the linear portions of the form, and shift back into the base character set (^O = control-O = decimal 15) to structure the alphanumeric portions of the form.

FORMS MODE (FORMAT MODE)

In Forms Mode, the terminal prevents you from overwriting or transmitting data in protected fields. Forms Mode is normally entered under control of the computer. Forms Mode is turned on by sending ESC W (the cursor is homed to the beginning of the first unprotected field). Normal operation is returned with ESC X (the cursor remains in its present position).

Protected Fields

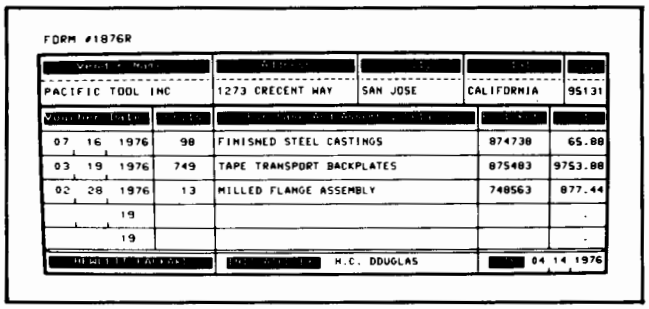
Fields can be protected so that displayed data cannot be overwritten or sent to a computer. When the terminal is placed in "Forms Mode" (Format Mode) all character positions on the screen are protected except those fields that have been specifically defined as "unprotected".

Unprotected Fields

Data can be written into unprotected fields in the normal manner. After reaching the end of an unprotected field, the cursor moves to the beginning of the next unprotected field.

The tab functions can be used to move from one unprotected field to the beginning of the next unprotected field. ESC i causes the cursor to be positioned at the beginning of the previous unprotected field. Fields are defined as "unprotected" by using ESC (at the start of the field. ESC) or the end of the line is used to end the field.

In the following figure, only the fields shown in white are unprotected. Even if the operator moves the cursor to a protected field and types a character, the cursor will move to the nearest unprotected field before displaying the character.



NOTE

Although the terminal does not support "transmit only" fields, if the "transmit only" escape sequence (ESC {) is sent from the computer, it is redefined as an unprotected field.

Printer Control

SECTION V

INTRODUCTION

The HP 2623A Graphics Terminal includes an asynchronous RS-232C port for connecting the terminal to an external printer. As an option, either the HP 2622A or the HP 2623A terminal may also include an integral printer.

With either or both printers present, you may do any of the following:

- Initiate and terminate print operations programatically from a host computer.
- Print the line containing the cursor.
- Print all lines from the one containing the cursor through the bottom line on the screen.
- Print all lines from the one containing the cursor through the end of display memory .
- Copy a configuration menu from the screen to either or both printers.
- Enable data logging (to occur either from the top or bottom of display memory, as designated by you when you enable it).
- Perform a line feed (advance the paper one line).
- Perform a form feed.
- Print the contents of graphics memory on the HP 2623A.

With the optional integral printer, you may also do any of the following:

- Select expanded, compressed, or normal-sized print characters.
- Select continuous forms mode, report format (60 text lines per page), or metric format (64 text lines per page).

All of the above printer control functions can be initiated either locally by operator keystrokes or remotely by escape sequences sent from a host computer.

SELECTING INTEGRAL PRINTER MODES

At power-on time or after a hard reset, the integral printer is automatically reset to print in continuous forms mode using normal-size characters (80 characters per line, 10 characters to the inch).

To enable or disable the various printer modes (expanded characters, compressed characters, report format, metric format, or data logging), you must get to the "device modes" set of system function keys. One way of doing so is the following keystroke sequence:

This changes the function key labels to the following:

The use of f2 - f8 "device modes" keys are described in the next few topics below.

To copy graphics data from graphics memory to the integral printer, you may either initiate the transfer from the graphics keypad or by an escape sequence. (SHIFT NUM toggles the function of the graphics/numeric pad.) Pressing the [GRAPH COPY] key on the graphics/numeric pad initiates the transfer. Copying graphics data under program control (using the Esc&p sequence) is discussed later.

Record Mode

Record Mode copies data from the display or datacomm to the selected "to" device(s), depending upon whether or not the terminal is in Remote Mode.

- If in local mode (not Remote), the contents of display memory (alphanumeric display memory in the HP 2623A) is copied to the selected "to" device.
- If in Remote Mode, the data stream on the datacomm line is sent directly to the selected "to" device.

Record Mode may be initiated from either the keyboard or from an Esc&p escape sequence. To initiate record mode from the keyboard, press

An asterisk will appear in the softkey label to indicate that record mode is enabled. While in record mode, the keyboard is disabled except for the BREAK, RESET, and "RECORD MODE" keys. Pressing RESET, or SHIFT, CTRL, RESET, or "RECORD MODE" softkey will terminate record mode.

To initiate record mode from an excape sequence, send:

Turn on Record Mode: ESC&p <NUM>p 20C
(<NUM>p is optional)

The optional <NUM> parameter defines the character which may be used to turn off record mode. <NUM> is the decimal equivalent of an ASCII character that will turn off record mode if it is the first character in a record. The default is "0". If "<NUM>p" is omitted, or if "0p" is specified, no character will terminate record mode. Termination can only occur by pressing the "RECORD MODE" softkey, pressing RESET or ESC g (a soft reset), or pressing CTRL SHIFT RESET or ESC E (a hard reset).

The termination character is valid only for the current activation of record mode (i.e., when record mode is terminated, the termination character returns to "0", the default character).

The terminal returns an "S", "F", or "U" to the host computer if the escape sequence is received from datacomm. An "S" means that the terminal executed the escape sequence successfully; an "F" means that the terminal failed to execute the escape sequence; a "U" means that you terminated execution. This could be done by pressing the "REMOTE MODE" softkey on the keyboard.

When the status character is sent depends upon whether or not DC1/DC2 handshake is enabled. (Refer to the Terminal Configuration Menu discussion in Section II for handshake types.) If DC1/DC2 handshake is disabled, the character is sent immediately after the escape sequence is received from the host computer. If the handshake is enabled, the character is sent after record mode is turned off and a DC1 is received from the host computer.

A 256-character buffer is used to hold each record prior to sending it to the specified "to" device(s). If the record exceeds 256 characters, the terminal's handshake holds off any further transmission from the host until the buffer's contents is sent to the "to" device(s). Records shorter than 256 characters are indicated by a LF (linefeed) character. Again, the terminal's handshake holds off any further transmission from the host until the record in the buffer is sent to the "to" device(s).

If record mode is turned off, the contents of a partially filled buffer will be sent to the "to" device(s).

If the record mode termination character is the first character into the buffer, record mode is terminated; the termination character is not sent to the "to" device(s).

Expanded Characters

The integral printer can print expanded characters in which each print line contains up to 40 characters spaced five to the inch (see figure 5-1).

From the keyboard, you enable and disable expanded character printing by pressing the EXPAND PRINT key (f5). When enabled, an asterisk appears in the key label.

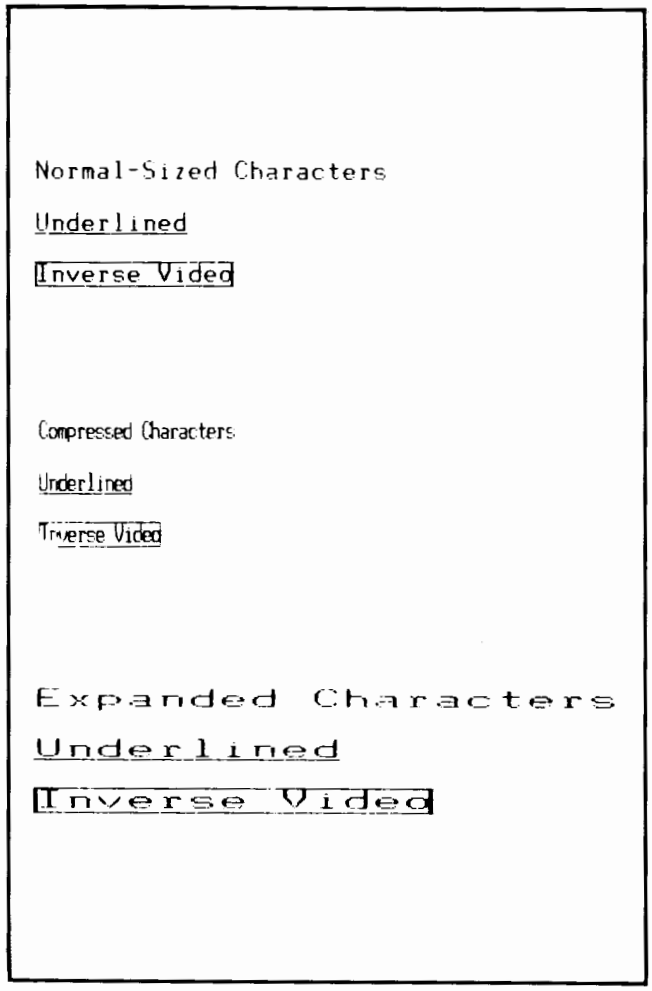


Figure 5-1. Character Sizes and Enhancements as Printed on the Integral Printer

If data is being copied from display and a line is longer than 40 characters, the printer automatically performs a CR LF after 40 characters and prints the remainder of the characters on the next line.

From a program operating in a host computer, you enable and disable the printing of expanded characters using the following escape sequences:

ENABLE: ESC&k 1S or ESC&p 15C
DISABLE: ESC&k 0S or ESC&p 14C

Once the printing of expanded characters is enabled, it remains enabled until explicitly disabled, until compressed characters are enabled, until a hard reset is performed, or until the power is turned off.

Compressed Characters

The integral printer can print compressed characters in which each print line contains up to 132 characters spaced 16.2 to the inch (see figure 5-1).

From the keyboard, you enable and disable compressed character printing using the COMPRESS key (16). When enabled, an asterisk appears in the key label.

From a program executing in a host computer, you enable and disable the printing of compressed characters using the following escape sequences:

ENABLE: Esc&k 2S or Esc&p 16C
DISABLE: Esc&k 0S or Esc&p 14C

Once the printing of compressed characters is enabled, it remains enabled until explicitly disabled, until expanded characters are enabled, until a hard reset is performed, or until the power is turned off.

Lines longer than 80 characters can only be sent to the printer from datacomm; lines from the display do not exceed 80 displayable characters. When sending lines from datacomm to the printer, either the Esc&pW escape sequence or Record Mode may be used. The method of using the escape sequence follows:

Define integral printer as the destination device: Esc&p 6D

Send records to the destination device:
Esc&pW (first record)
Esc&pW (second record)
.
.
.
Esc&pW (last record)

To send data to the printer using record mode, type the appropriate host computer system command to list the data; then, before terminating the command, enter record mode. You can do this from the keyboard by pressing:

[RIDS], device control, device modes, RECORD MODE *

After record mode is entered, terminate the command (usually this is done by pressing RETURN). Data received over datacomm is sent directly to the printer; it does not appear on the display. Record mode may be turned off by pressing the "RECORD MODE" softkey again (the asterisk will disappear from the key label).

The details of record mode are discussed previously in this section.

Report Format

The integral printer normally operates in continuous forms mode without regard for page boundaries. You can, however, enable report format in which printed output is treated as a series of 66-line pages (a 3-line top margin, 60 lines of text, and a 3-line bottom margin). The margins and text area together form an 8½ inch by 11 inch page. The printer uses a small tic mark to mark the end of one page and the beginning of the next. Report forms mode is shown in figure 5-2.

From the keyboard, you enable and disable report format using the REPORT PRINT key (f7). This key alternately enables and disables report format. When enabled, an asterisk appears in the key display; the asterisk in f8, if on, is turned off; and the printer skips 3 lines, prints a page break, and skips 3 lines.

From a program executing in a host computer, you enable and disable report format using the following escape sequences:

ENABLE: Esc&p 17C
DISABLE: Esc&p 19C

Once enabled, report format remains enabled until explicitly disabled, until metric format is enabled, until a hard reset is performed, or until the power is turned off.

Metric Format

The integral printer normally operates in continuous forms mode without regard for page boundaries. You can, however, enable metric format in which printed output

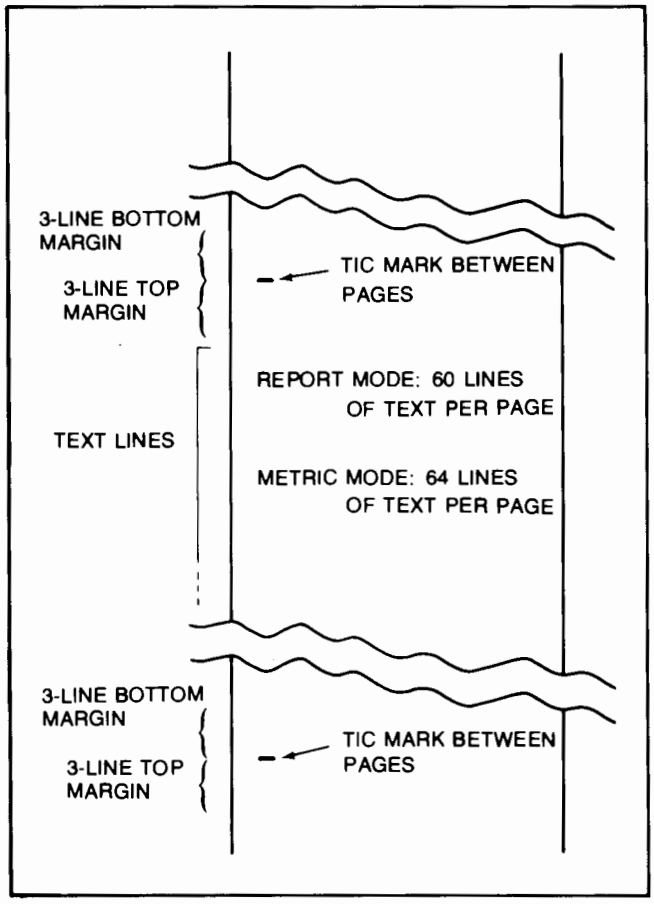


Figure 5-2. Report and Metric Formats

is treated as a series of 70-line pages (a 3-line top margin, 64 lines of text, and a 3-line bottom margin). The printer uses a small tic mark to mark the end of one page and the beginning of the next. Metric format is illustrated in figure 5-2.

From the keyboard, you enable and disable metric format using the METRIC PRINT key (f8). This key alternately enables and disables metric format. When enabled, an asterisk appears in the key display; the asterisk in f7, if on, is turned off; and the printer skips 3 lines, prints a page break, and skips 3 lines.

From a program executing in a host computer, you enable and disable metric format using the following escape sequences:

ENABLE: Esc&p 18C
DISABLE: Esc&p 19C

Once enabled, metric format remains enabled until explicitly disabled, until report format is enabled, until a hard reset is performed, or until the power is turned off.

DATA LOGGING

The terminal includes a mechanism called "data logging" whereby data can be automatically routed to the integral printer and/or an external printer. There are two types of data logging: top and bottom.

Top Logging

When the display is filled and another line of data is entered through the keyboard or received over a datacomm line, the top line in the display is purged to make room for the new line. With top logging, each line that is purged from the top of the display is printed. Thus, while the line is "lost" from display memory, it is maintained in hard copy form.

Bottom Logging

With bottom logging, each time the cursor moves from one line to another as the result of an explicit line feed or an end-of-line-wraparound, the line from which the cursor moved is printed. This feature allows you to maintain a hard copy "trail" of all lines added to the display in the order in which they were entered and/or received.

When performing data logging in remote mode, the terminal and host computer must be using the ENQ-ACK or XON-XOFF handshakes or they must be using a baud rate that is equal to or less than the rate at which the slowest selected printer can function. (For individual lines being logged sporadically on the integral printer, 1200 baud may work; for a series of successive lines, however, you will probably have to drop to 600 baud.)

From the keyboard, you enable and disable data logging using the LOG TOP (f4) and LOG BOTTOM (f3) keys. These keys alternately enable and disable top logging and bottom logging, respectively. When either is enabled, an asterisk appears in the associated key display.

From a program executing in a host computer, you enable and disable data logging using the following escape sequences:

ENABLE BOTTOM LOGGING: Esc&p 11C
ENABLE TOP LOGGING: Esc&p 12C
DISABLE LOGGING: Esc&p 13C

Both forms of data logging may NOT be enabled simultaneously.

Once either form of data logging is enabled, it remains enabled until explicitly disabled, until the other form of data logging is enabled, until a hard reset is performed, or until the power is turned off.

Note that the keyboard is temporarily locked while a line of data is being "logged". This may make it difficult to perform any keyboard operations if a large quantity of data is coming into the display over a datacomm line rapidly enough to result in continuous logging.

DISPLAY TO PRINTER ALPHANUMERIC DATA TRANSFERS

The display is defined as the "from" device in alphanumeric data transfers to the integral printer and/or external printer. If you define the integral printer and/or external printer as destination ("to") devices, you can use the "device control" set of system function keys to print one or more lines of data from the display. One way of getting to the "device control" keys is the following keystroke sequence:

[AIDS] [f1] device control

This changes the function key labels to the following:

Press f3 for the integral printer on the HP 2622A.

For data transfers initiated through the keyboard, display memory is always the source ("from") device while the integral printer and/or an external printer are the destination ("to") devices. To select an external printer as the "to" device, press the "TO EXT DEV" function key (f2). To select the integral printer as the "to" device, press the "TO INT PRT" function key (f3). When a printer is selected as the current "to" device, an asterisk appears in the associated function key label. You may, if you so desire, have both printers selected as "to" devices simultaneously.

Programmatically, you can define the "to" devices by using a device control escape sequence (Esc&p):

Select external device:	Esc&p 4D
Select internal printer:	Esc&p 6D
Select both:	Esc&p 4d6D

Copy Line

When either or both of the printers are selected as a destination device, you can copy the line containing the cursor from the display to the printer(s). The entire line is copied. Block terminators are ignored. After the line is printed, the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin). If the cursor is on an empty line, COPY LINE should not cause anything to print.

From the keyboard, you copy one line of data using the COPY LINE key (f8) in the device control set of system function keys.

From a program executing in a host computer, you copy one line of data using one of the following escape sequences:

Esc&p B or Esc&p 0B

Copy Page

When either or both of the printers are selected as the destination device, you can copy all lines, starting with the line containing the cursor through the last line visible on the screen, to the printer(s). Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, the printer does nothing.

From the keyboard, you copy a page of data using the COPY PAGE key (f2) in the device control set of system function keys.

From a program operating in a host computer, you copy a page of data using one of the following escape sequences:

Esc&p F or Esc&p 0F

Copy All

When either or both of the printers are selected as the destination device, you can copy all lines, starting with the line containing the cursor through the last line of display memory, to the printer(s). Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin). If the cursor is at a line that is beyond the last displayable line, the printer does nothing.

From the keyboard, you copy "all" using the COPY ALL (f6) key in the device control set of system function keys.

From a program executing in a host computer, you copy "all" using one of the following escape sequences:

Esc&p M or Esc&p 0M

Copy All of Display Memory

When either or both of the printers are selected as a destination ("to") device, you can copy all of display memory to it by using an Esc0 sequence. In response to this sequence, the terminal homes the cursor and then copies the entire content of display memory to the printer(s).

During the data transfer, block terminators and non-displaying terminators within display memory are ignored.

The Esc0 sequence may be entered through the keyboard, issued from a user-defined function key, or issued from a program executing in a host computer.

When the terminal is in local mode, pressing the ENTER key performs this same function.

Copy Menu

When either or both of the printers are selected as a destination ("to") device and a configuration menu is currently being displayed on the screen, you can copy the menu to the printer(s) by pressing the ENTER key.

Skip Line

When either or both of the printers are selected as a destination device, pressing the ADVANCE LINE key (f5) in the device control set of system function keys sends an ASCII CR LF control code sequence to the printer(s), thus causing the paper to be advanced by one line.

Programmatically, you can cause a line feed on the integral printer by using the following device control escape sequence:

INTEGRAL PRINTER:	Esc&p 1c 6u 1P
EXTERNAL PRINTER:	Esc&p 1c 4u 1P

The "p" parameter in the above escape sequences specifies how many line feeds you wish performed. To initiate four successive line feeds, for example, merely substitute "4P" for the "1P" sequence.

Skip Page

When either or both of the printers are selected as a destination device, pressing the ADVANCE PAGE key (f4) in the device control set of system function keys sends an ASCII ^F control code to the printer(s), thus causing the paper to be advanced to the top of the next page.

When the integral printer is selected as a "to" device, this control function causes a true form feed only if report format or metric format is enabled. In all other cases, the advance page function merely causes the integral printer to advance the paper one line.

Programmatically, you can cause a form feed on the printer by using the following device control escape sequence:

INTEGRAL PRINTER: ^[&p 0c 6U
EXTERNAL PRINTER: ^[&p 0c 4U

Note that the values 2 through 10 may also be used with the "c" parameter (instead of the zero); this will also initiate one form feed.

Also note that the parameter "4U" may be redefined by way of the PrinterCode4 configuration parameter to signify the integral printer in the HP 2623A (see description of the External Device Configuration menu discussed later in this section.)

The HP 2622A terminal directs "4U" controls to "6U" (the optional integral printer).

The control code for a Form Feed (CNTRL L) produces various results depending on the conditions existing when a Form Feed is encountered. If REPORT mode is disabled, Form Feed is ignored.

If REPORT mode is enabled, and LOG TOP and LOG BOTTOM are disabled, Form Feed results in a top-of-page operation on the printer (skip to new page and print tic mark).

If REPORT mode is enabled, and LOG TOP is enabled, Form Feed results in completion of logging from the screen to the printer, a top-of-page operation on the printer (skip to new page and print tic mark), and a top-of-page operation on the screen (home up, clear display).

If REPORT mode is enabled, and LOG BOTTOM is enabled, Form Feed results in a top-of-page operation on the printer (skip to new page and print tic mark), and a top-of-page operation on the screen (home up, clear display).

Device Control Completion Codes

After issuing a copy line, copy page, copy all, skip line, or skip page ^[&p sequence, the remote program determines whether or not the operation was successfully performed by executing an INPUT or similar instruction that requests one ASCII character from the terminal. The terminal responds by sending an "S", "F", or "U". An "S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the terminal operator interrupted the data transfer by pressing RETURN . Note that these completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled ("locked") until the status is sent.

Note that in either character or block line mode, the terminal sends a ^M (or a ^M ^J if auto line feed mode is enabled) following the completion code. In block page mode, it sends a block terminator character (as defined in the Terminal Configuration menu described in Section II of this manual).

If a datacomm error occurs during the transmission of the data record, the device control completion code is unpredictable. Datacomm errors are reported by way of the terminal status bytes described in Section VII.

GRAPHICS MEMORY TO PRINTER DATA TRANSFERS (HP 2623A)

Graphics memory to printer data transfers may be initiated either by the graphics/numeric keypad or by escape sequence. The [GRAPH COPY] key initiates the transfer from the keypad. Note that the graphics functions of the keypad are enabled at power-on, hard reset, or when toggled by the SHIFT NUM keys. The escape sequence to initiate the transfer is:

Initiate graphics transfer: ^[&p7sF

Note: 7s sets up the source as graphics memory, the default is alphanumeric memory.

The integral printer is the default destination device. An external printer may be selected as the destination by the SIDS system softkeys or ^[&p escape sequence. The external printer is configured by the External Device Configuration Menu (discussed later in the section). When selecting the external printer as the destination for graphics data, parity must be set to None in the configuration menu.

To select the external printer by using the softkeys, press

[AIDS], device control [f1], to devices [f3]

This changes the function key labels to the following:

Press f2 for the external printer. An asterisk appears in the label when a device is selected.

To select either or both printers as the destination by using the escape sequence:

Select external printer as the destination:	Esc&p 4D
Select integral printer as the destination:	Esc&p 6D
Select both printers as the destination:	Esc&p 4d 6D

You may combine device selection and transfer initiation in one escape sequence:

Select external printer as destination and initiate graphics copy:	Esc&p 4d F

COMPUTER TO TERMINAL DATA TRANSFERS

When either of both of the printers are selected as a destination ("to") device, you can initiate a data transfer from a program executing in a host computer to the printer(s) by using the following device control escape sequence:

Esc&p <character-count> W <record>

where:

<character-count>	is an integer within the range 1-256 specifying the number of binary bytes in <record>. The record is terminated when the specified number of binary bytes have been transmitted. If this parameter is not present ASCII transfers are initiated and the record is terminated when the 256th data byte after the "W" is transmitted or by the first ASCII LF code, whichever occurs first. If the record is terminated by an LF, the LF is also passed to the printer.
<record>	is the data record to be transmitted.

Example: Send the next 15 binary bytes from the computer to all "to" devices.

Esc&p 15W

This escape sequence is recognized only when received over a data comm line. It is ignored if entered through the keyboard.

You may include the desired destination device assignment(s) within the escape sequence by using the "d" command parameter. You may also, prior to issuing the above escape sequence, define the desired destination devices either locally through the keyboard or programmatically by way of a separate device control (Esc&p) sequence. In any case, the only destination devices that are recognized by this type of data transfer operation are the display (3d) and the integral printer (6d), and/or an external printer (4d).

If no destination devices are specified within the above escape sequence, the the current "to" device assignments are used. If nothing is currently selected as a "to" device, the data record is accepted over the data comm port and then is discarded by the terminal (also an "F" is returned as the device control completion code).

Binary transfers are of the form Esc&p <character-count> W <record>. ASCII transfers are of the form Esc&p W <record>, where as ASCII LF or the 256th data byte terminates the record. In binary transfers, all eight bits received are passed to the printer. Parity checking and transmission is disabled. In ASCII transfers, seven bits will be passed to the printer.

If the escape sequence does NOT include a <character-count>, then the following applies:

- If EnqAck = YES in the active data communications configuration menu, the data comm firmware strips all <ENQ> codes from the incoming data and responds to each by transmitting an <ACK>.

If the escape sequence includes a <character-count>, then the following apply:

- If EnqAck = YES in the active data communications configuration menu, an <ENQ> code immediately following the "W" and preceding the data record is treated as part of an Enq-Ack handshake (the data comm firmware strips the <ENQ> code from the incoming data and responds to it by sending an <ACK>).

- After the optional leading Enq-Ack handshake, ALL characters received are treated as data (including <ENQ>, <ACK>, <NULL>, and <DEL>) regardless of the setting of the EnqAck configuration fields.

When transferring a data record from the host computer to the printer using the above device control escape sequence, the remote program determines whether or not the operation was successfully performed by executing an INPUT or similar instruction that requests one ASCII character from the terminal. The terminal responds by sending an "S", "F", or "U". An "S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the terminal operator interrupted the data transfer by pressing RETURN. Note that these completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled ("locked") until the status is sent.

Note that in either character or block line mode, the terminal sends a CR (or a CR LF if auto line feed mode is enabled) following the completion code; in block page mode, it sends a block terminator character (as defined in the Terminal Configuration menu).

If a data comm error occurs during the transmission of the data record, the device control completion code is unpredictable. Data comm errors are reported by way of the terminal status bytes described in Section VIII of this manual.

INTEGRAL PRINTER SELF-TEST

The terminal includes a printer self-test feature that exercises the integral printer to verify that it is functioning properly.

From the keyboard, you initiate the printer self-test using the following keystroke sequence:

If the printer is present and functioning properly, it generates the test pattern shown in figure 5-3.

If an error condition is detected while the test is being executed, the message "INTEGRAL PRINTER ERROR" appears across the bottom of the screen. To clear the message, press RETURN. Note that the error condition may be either of the following, in which case you could correct it yourself:

1. Out of paper.
2. The metal latch (under the plastic printer lid) is not pressed down securely. See Section IX, figure 9-3.

The printer self-test cannot be initiated programmatically.

Note that while the printer self-test is in progress the terminal's interrupt mechanism is disabled. If you initiate this test while data is being received over a datacomm port, some of that data could get lost.

CONFIGURING THE EXTERNAL PRINTER

The HP 2623A has an external printer port for interfacing RS-232C serial printers. Configuring an external printer consists of cabling it to the port and specifying parameters in the configuration menu.

Cabling

The HP 2623A has an external device port, along with the data communications port, at the rear of the terminal (see figure 5-4).

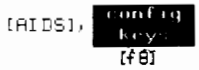
The HP 13242 cables listed in table 5-1 both have a male RS-232C connector on one end and either a male or female RS-232C connector on the other. The male end attaches to the external device port on the rear of the terminal, and the other end attaches to the external printer as shown in figure 5-5. For the HP 13242G Cable, which has a male connector on each end, it makes no difference which end is attached to the terminal.

Since the external device connector on the terminal is a standard RS-232C female connector, you can use cables other than those listed in table 5-1 as long as they have a male RS-232C connector on one end and their pin-outs are compatible with those of the HP 13242 cables.

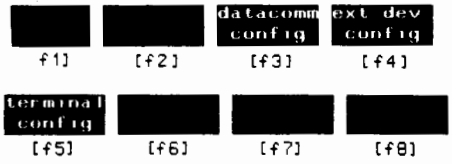
Filling In The Configuration Menu

Now that you have made the physical connections between the terminal and the external printer, you are ready to configure the terminal's external device port.

To configure the port, first use the following keystroke sequence:



This changes the function key labels to the following:



The **ext dev config** function key, when pressed, causes the external device configuration menu to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu (see figure 5-6). The configuration menu displays the currently stored menu parameters from non-volatile memory.

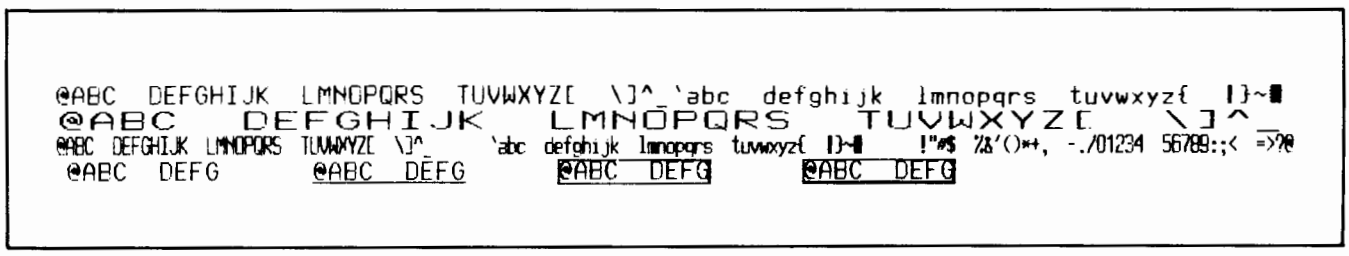


Figure 5-3. Integral Printer Self-Test Output

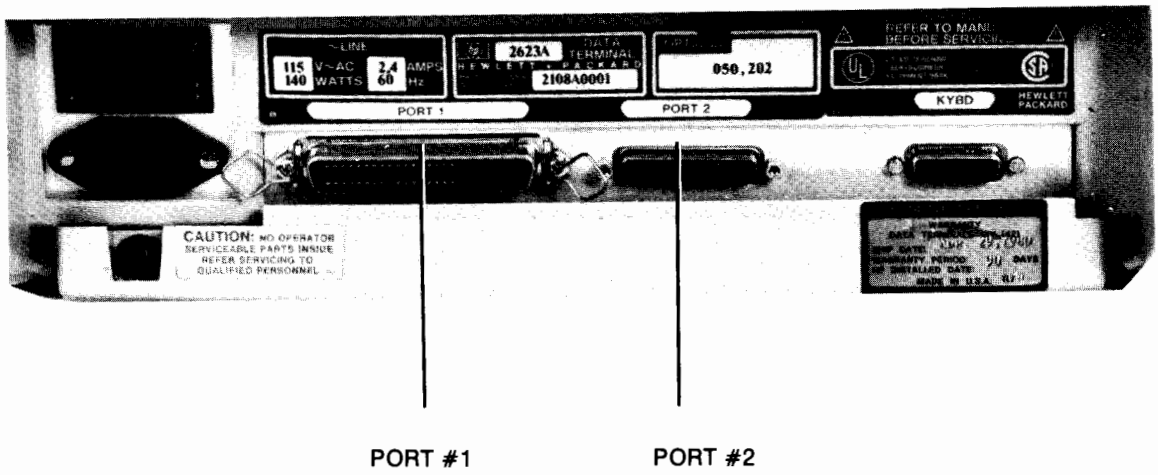


Figure 5-4. HP 2623A Graphics Terminal, Rear View

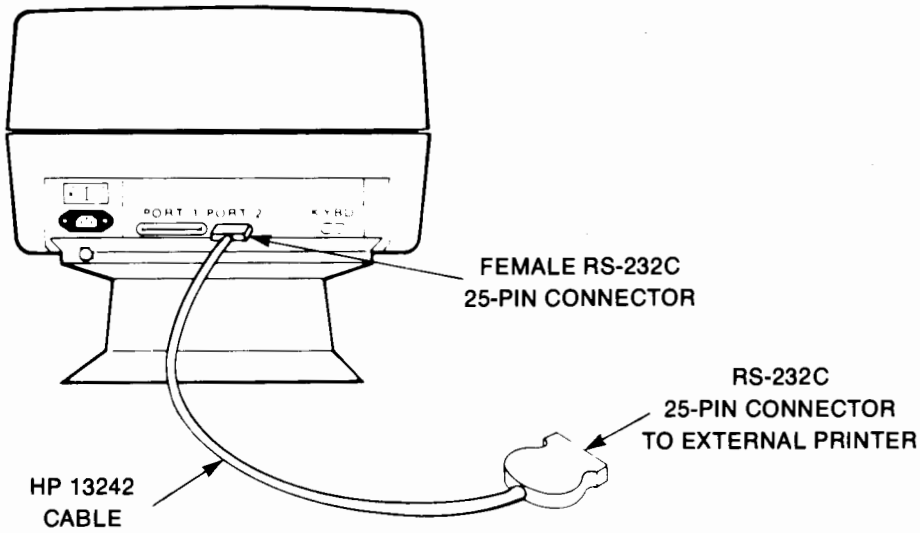


Figure 5-5. External Device Port Cabling (HP 13242 Cables)

Table 5-1. External Device Port Data Communications Cables

Cable No.	HP Part No.	Description
13242G	13242-60010	RS232 PRINTER CBL (MALE) Male RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2631 and HP 2635. Length: 15 feet (4.5 meters)
13242H	13242-60011	RS232 PRINTER CBL (FEMALE) Female RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers. Length: 15 feet (4.5 meters)

Whenever a configuration menu is displayed on the screen, the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the tab key. For most of the fields (the ones containing the underlined video enhancement) you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

The meanings of the various fields are described in table 5-2.

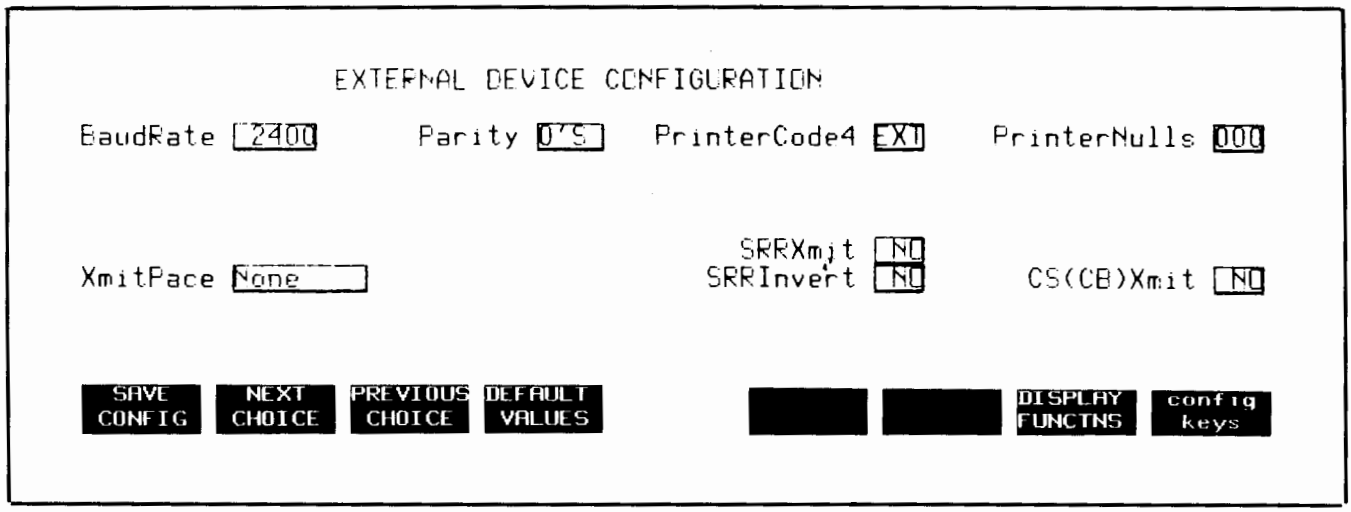


Figure 5-6. External Device Configuration Menu

Table 5-2. External Device Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600
Parity	This field specifies what type of parity generation and checking you wish used with each data character. (Note that parity must be set to None for binary transfers.) Values: NONE (no parity bit); 0'S (parity bit always zero); ODD (odd parity); 1'S (parity bit always one); EVEN (even parity)

Table 5-2. External Device Configuration Menu Fields (Continued)

PrinterCode4	The field specifies which printer (an external printer or the integral printer) will respond to device code "4" when the terminal receives a device control escape sequence from the host computer. Device code "4" is ordinarily used only for selecting an external printer. Through the use of this configuration parameter, however, you can redirect the device control operations to the integral printer without altering the host computer program. Values: EXT (external printer) INT (integral printer) Default: EXT
PrinterNulls	This field specifies the number of ASCII null codes (0–255) to be transmitted to an external printer after each ASCII control code. Value: 0-255 Default: 0
XmitPace	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (ASCII <DC3>), it stops transmitting data. When the terminal subsequently receives an XON code (ASCII <DC1>), it resumes transmitting data. This should not be used with DC1/DC2 handshaking. If this field is set to "NONE", the terminal does NOT recognize the ASCII <DC1> and <DC3> codes as XON and XOFF. For other forms of transmit pacing, refer to the descriptions of the SRRXmit and CS(CB)Xmit fields below. Values: NONE XonXoff Default: NONE
SRRXmit	This field specifies whether or not a true state (−12V) on the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is a required condition for transmitting data. This mechanism is primarily used in conjunction with printers which must be able to control the transmission of data from other devices. The SRR/SCF control line is connected to RS-232C pin number 12. Values: YES NO Default: NO
SRRInvert	This field applies only when the SRRXmit field is set to "YES". When both the SRRxmit and SRRInvert fields are set to "YES", the true state of the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is inverted from −12V to +12V. Values: YES NO Default: NO
CS(CB)Xmit	This field specifies whether or not a true state (−12V) on the RS-232C Clear to Send (CS/CB) control line is a required condition for transmitting data. For a modem configuration, it is set to "YES". Also, if the Asterisk field is set to "CS" then CS(CB)Xmit should be set to "YES". Values: YES NO Default: NO

Data Communications

SECTION VI

INTRODUCTION

The term "data communications" (or "datacomm") refers to the transfer of data between the terminal and a host computer.

There are several ways to connect the terminal to a computer. To arrive at a particular way you must compare a number of factors and make a series of decisions. After selecting the necessary equipment and cables, you must then physically connect the terminal to the computer (or to the modem, if that is what you have chosen) and configure the terminal for use with the particular type of data communications link.

This section is divided into four parts:

1. The first is a general discussion that should help you decide what type of equipment and cabling you need for the data link.
2. The second tells you how to physically install the terminal.
3. The third tells you how to configure the terminal to operate properly with the selected type of data link.
4. The final part provides programming reference material for someone who is writing a datacomm driver or controller program to communicate with the terminal in a point-to-point environment.

Before proceeding with the decision making process, it may help to briefly define the most important terminology as it pertains to data communications.

Communications Link: The means by which a terminal is connected to a host computer. This always includes some type of communications line (a coaxial cable, the public telephone network, or a leased telephone line), and it may also include a pair of modems (one at each end of the line).

Point-to-Point: A data communications configuration in which a single terminal is connected to a host computer over a communications link. The terminal is designed for use with a point-to-point communications link.

Asynchronous: A mode of transmission in which each data character is framed by a "start bit" and one or more "stop bits". The interval between successive data characters is random. The terminal is designed for use with an asynchronous communications link.

Full Duplex: A communications link in which data can be transmitted in both directions simultaneously. The terminal is designed for full duplex data communications.

Character Mode: When the terminal is operating in character mode, it sends data characters to the computer one at a time as they are typed into the keyboard.

Block Mode: When the terminal is operating in block mode, data characters typed into the keyboard are merely stored in display memory. When a block transfer is subsequently triggered (by the host computer or by pressing the [ENTER] key), a group of data characters is sent from the terminal to the computer as a block.

A point-to-point configuration is the standard form of data communications within the industry (it is sometimes referred to as a "Teletype-compatible" communications link). Point-to-point is supported by most computers. At any given time, it accommodates only one terminal per communications link; it may, however, operate in either character mode or block mode.

A major consideration in selecting which type of connection to use is the anticipated distance between the terminal and the computer. If the terminal will be located in the vicinity of the computer system, you may use a hardwired connection. The Electrical Industry Association (EIA) Standard RS-232-C (to which the terminal conforms) limits cable lengths to a maximum of 15 meters (50 feet).

Another consideration is the desired availability of the particular computer port. If you wish to have it available (at different times) to terminals in diverse and/or varying locations, then you should choose a modem connection with dial-up capability.

HARDWIRED CONNECTIONS

If you decide on a point-to-point hardwired connection, the only decision that remains to be made is the type of cable to be used. The available cables are summarized in table 6-1. Please note that a hardwired connection for your terminal is always full duplex (the terminal does not support half-duplex connections).

MODEM CONNECTIONS

If you decide on a point-to-point modem connection, you must now decide what type of modem to use. Note that point-to-point as supported by the terminal always employs asynchronous transmission. You will therefore be limiting your choice of modem to the asynchronous variety. Refer to table 6-2 for help in selecting the proper modem.

Table 6-1. Data Communications Cables and Test Connectors

Cable No.	HP Part No.	Description
13222C	13222-60003	TERMINAL TO RS232 CABLE 50 PIN TO 25 PIN CABLE Female RS-232-C 25-pin connector. Length: 2 meters (6.6 feet)
13222M	13222-60002	EUROPEAN MODEM CABLE Male RS-232-C 25-pin connector for interfacing the terminal to the European telephone system via Bell 103 or 202C type European modems. Length: 5 meters (16.7 feet)
13222N	13222-60001	U.S. MODEM CABLE Male RS-232-C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexer; to a Bell 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only). Length: 5 meters (16.7 feet)
13222W	13222-60007	13222-60007 (W) Female RS-232-C 25-pin connector for interfacing the terminal to an HP 300 Computer System. Length: 5 meters (16.7 feet)
13222Y	13222-60005	EMP PROTECT (MALE) Male RS-232-C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 5 meters (16.7 feet)
13232U	5061-2403	Modem bypass cable with a female RS-232-C 25-pin connector on both ends. It crosses the signals so that two terminals can communicate with one another. Length: 1.5 meters (5 feet)
—	02620-60056	This is a male 50-pin test hood for use on port #1.
—	02620-60062	This is a male RS-232C test hood for use on port #2.
—	02645-6004	This is a female RS-232C test hood for use on an HP 13222 or 13242 datacomm cable. They are available as a 3-item set which can be ordered either with the terminal (as option 981) or separately (as the HP 13259A Datacomm Self-Test Connector Kit).

Modem Considerations

If you are communicating with the host computer through a modem, it may be necessary for you to turn on a modem power switch or make modem parity setting changes. The modem's baud rate and parity settings should be the same as those configured in the terminal.

The terminal supports the Bell 103A, Bell 212A, or equivalent type of modem.

Whenever the modem line (Data Set Ready) is active, an asterisk appears between the fourth and fifth screen labels at the bottom of the screen. If your facility requires the display of this "active modem" indicator, do not shut off the screen labels display.

The asterisk between the fourth and fifth screen label is controlled by an LED which tracks the Data Set Ready (DSR) input line to the terminal. When a modem is connected, the DSR line is low (active) and the modem indicator (asterisk) is on. When the DSR line is high, this signals a modem disconnect and the asterisk disappears from the screen.

INSTALLING A POINT-TO-POINT CONFIGURATION

The terminal's datacomm port may be connected to a computer via a 50-pin, female RS-232-C compatible connector provided on the back of the terminal (see figure 6-1).

Table 6-2. Modems

MODEM	Data Rate (Bits/Sec)	Duplex Full/Half	Dialed/ Leased	Reverse Channel
HP 13265A	300	F	D	No
Bell 103A	300	F/H	D/L	No
Bell 202T Bell 202D	1200 (see note 2)	F/H	L	Option
Vadic VA3400 (see note 1)	1200	F	D	No

NOTES: 1. Can be configured for either asynchronous or synchronous operation. With the HP 2622A or HP 2623A, however, it must be configured for asynchronous operation.
2. C2 line conditioning allows operation at 1800 bits per second.

CABLING

The HP 13222 cables listed in table 6-1 all have a male 50-pin connector on one end and either a male or female RS-232-C connector on the other. The 50-pin end is the wider of the two (approximately 7 cm or 2¾ inches wide) and you attach it to the connector on the rear panel of the terminal. The RS-232-C end attaches to the modem, computer multiplexer panel, external printer, or interface cable as illustrated in figure 6-2.

You may also connect either an HP 13265A Modem or an HP 13266A Current Loop Converter to the connector as illustrated in figure 6-3.

CONFIGURING THE TERMINAL

Once the physical connections between the terminal and the computer or modem are complete, the terminal can be configured.

To configure the datacommunications portion of the terminal, first use the following keystroke sequence:

[AIDS], [f8] config keys, [f3] datacomm config

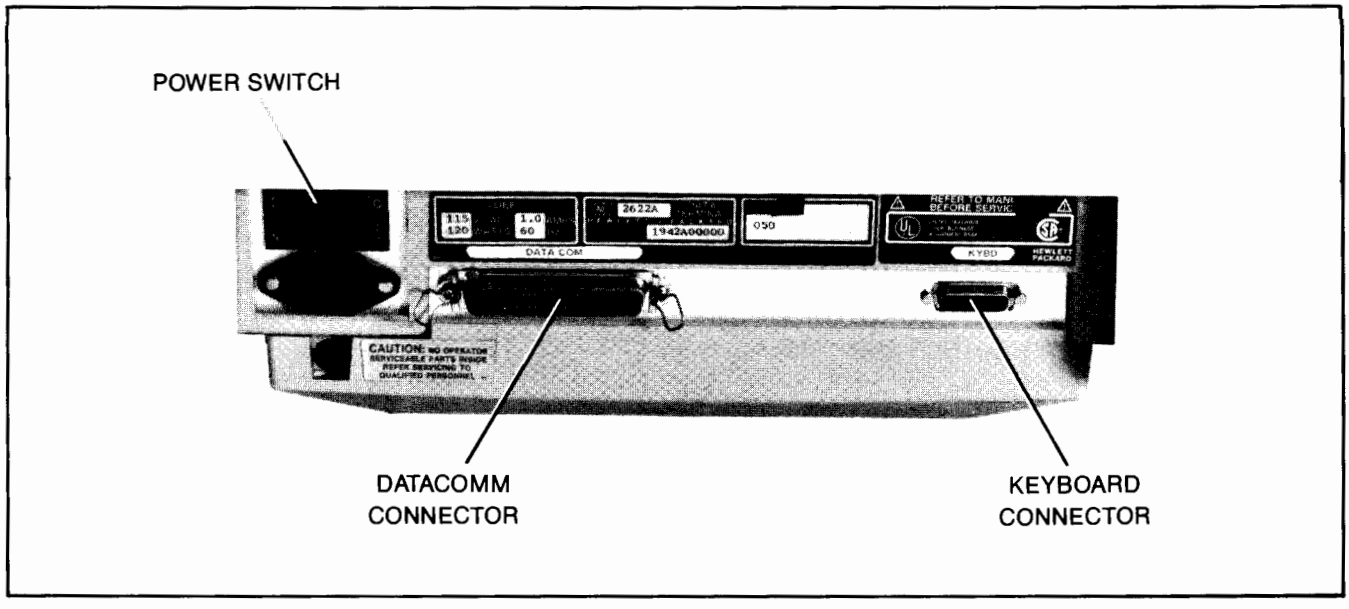


Figure 6-1. HP 2622A Display Terminal, Rear View

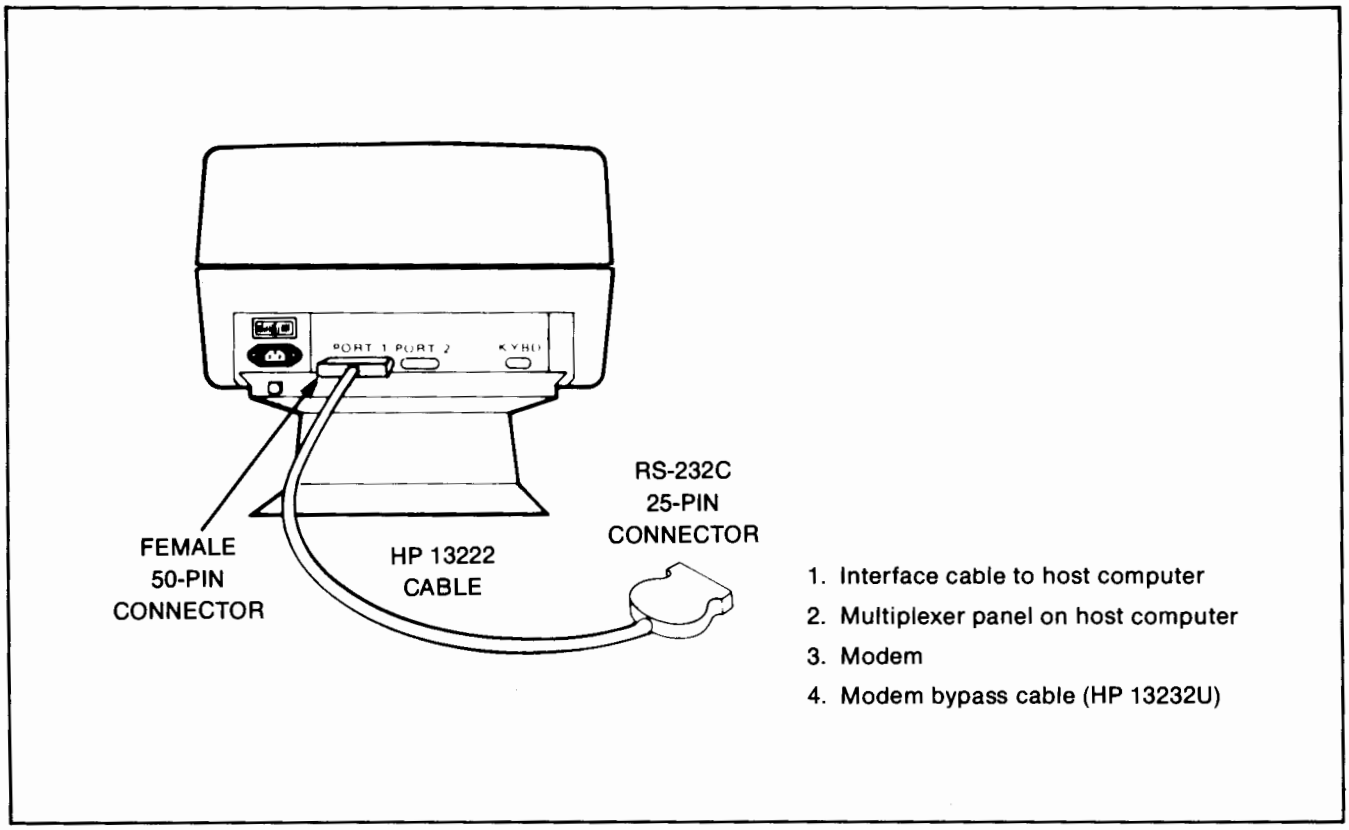


Figure 6-2. HP 2623A Terminal Cabling (HP 13222 Cables)

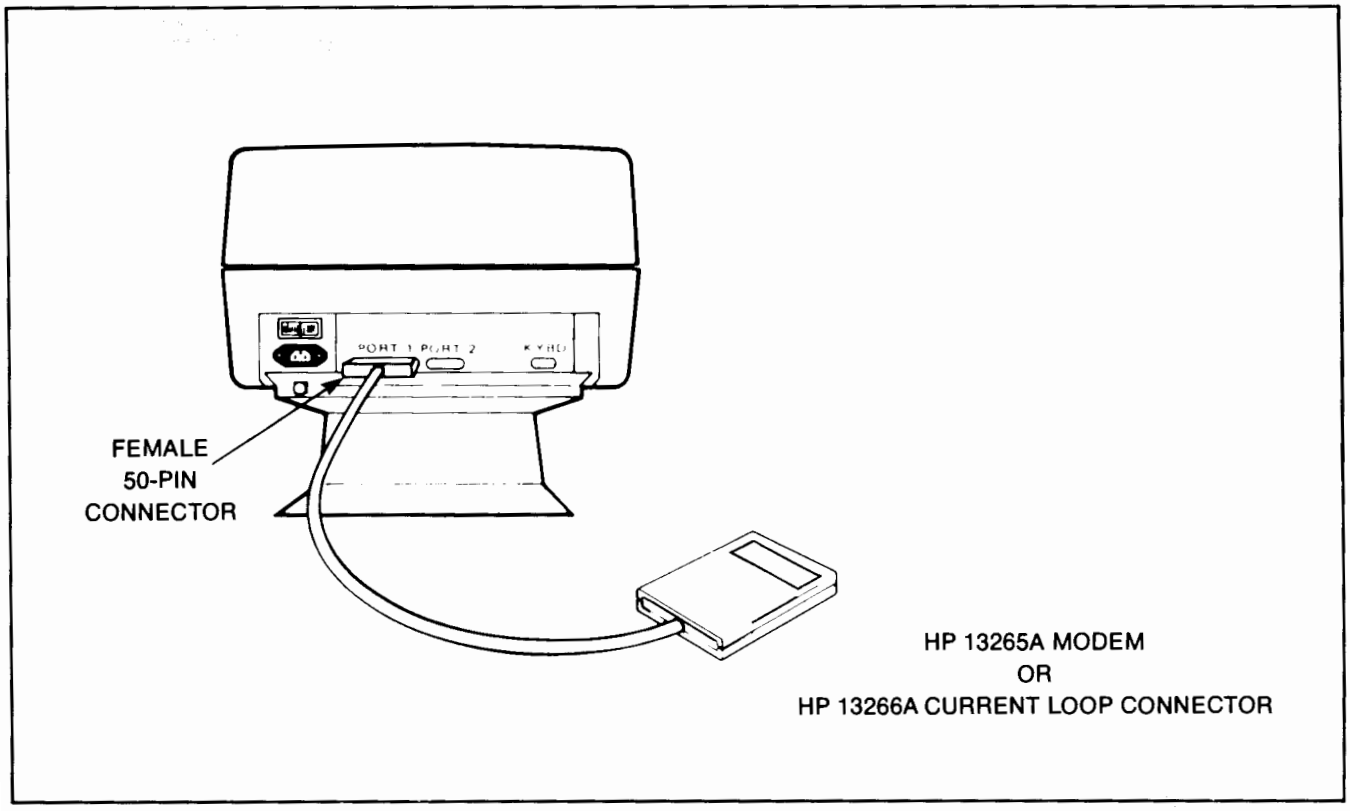


Figure 6-3. HP 2623A Terminal Cabling (HP 13265A Modem or HP 13266A Current Loop Converter)

When you press the **datacomm config** (f3) function key, the datacomm configuration menu currently stored in non-volatile memory appears on the screen and the function key labels change to the following:

[f1]	[f2]	[f3]	[f4]
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES
[f5]	[f6]	[f7]	[f8]
		DISPLAY FUNCTNS	config keys

Note that if you have not previously stored a menu in non-volatile memory, the default values are displayed in the configuration menu (see figure 6-4).

The menu contains a set of unprotected fields that you access using the TAB key. You select the desired parameters in these fields using the NEXT CHOICE (f2) and PREVIOUS CHOICE (f3) function keys.

The meanings of the various fields in datacomm menu are described in table 6-3.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the SAVE CONFIG (f1) function key. Note that when you do this, the particular datacomm configuration takes effect immediately.

While the datacomm configuration menu is displayed on the screen, the f4 and f8 keys have the effects described below:

[f4] DEFAULT VALUES	Pressing this key causes all fields in the menu to be filled with their default values.

```
                    DATACOMM CONFIGURATION

BaudRate 2400      Parity 0'S                    EnqAck YES

                   Chk Parity NO      SR(CH) LO

RecvPace None                                    CS(CB)Xmit NO

XmitPace None
```

Figure 6-4. Datacomm Configuration Menu

Table 6-3. Datacomm Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place. Values: 110, 134.5, 150, 300, 600, 1200, 1800, 2400 (default), 4800, 9600 **NOTE** For 110 baud, the terminal is automatically configured to transmit 2 stop bits with the data to the computer. At 110 baud, the terminal also expects to receive 2 stop bits with the data received from the computer. For all other baud rates, 1 stop bit is transmitted with data and expected to be received with data.
Parity	This field specifies what type of parity generation and checking you wish used with each data character. Values: NONE (no parity bit) 0'S (parity bit always zero) (default) ODD (odd parity) 1'S (parity bit always one) EVEN (even parity)
EnqAck	This field enables or disables the use of the Hewlett-Packard ENQ-ACK handshake. This type of handshaking is described under "Pacing Mechanisms" in the "Point-to-Point Programming Information" portion of this section. Values: YES (enable) (default) NO (disable)

Table 6-3. Datacomm Configuration Menu Fields (Continued)

Chk Parity	This field is used for enabling or disabling the parity check feature for data characters received over the datacomm line. Note that if the Parity field (above) is set to NONE, then this field is ignored. Note that parity is still transmitted. Values: YES (enable) NO (disable) (default)
SR(CH)	This field specifies the desired state of the RS-232-C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS-232-C pin number 23, is defined as the Data Signal Rate Detector (DTE Source). It is normally used on dual speed modems to select the appropriate speed (single speed modems merely ignore this line). Values: HI LO (default)
RecvPace	Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. There is one way of performing receive pacing: by using the XON and XOFF control codes. If this field is set to "XonXoff", the terminal will automatically perform receive pacing using XON (ASCII $^{Q}_{1}$) and XOFF (ASCII $^{S}_{3}$) control codes. With this type of receive pacing, the terminal causes the remote device to halt transmission by sending an XOFF code and to resume transmission by sending an XON code. For this type of receive pacing to work, the remote device must of course be configured to start and stop transmission in response to XON and XOFF codes. Note that if the remote device recognizes XON and XOFF codes and your terminal is operating in character mode, you can issue the codes through the keyboard regardless of the setting of this field. The CTRL and Q keys (when pressed simultaneously) generate an XON code and the CTRL and S keys generate XOFF. Values: NONE (default) XON/XOFF
CS(CB)Xmit	This field specifies whether or not a true state (−12V) on the RS-232-C Clear to Send (CS/CB) control line is a required condition for transmitting data. For a modem configuration, it is recommended that you set this field to "YES". Values: YES NO (default)
XmitPace	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (ASCII $^{S}_{3}$), it stops transmitting data. When the terminal subsequently receives an XON code (ASCII $^{Q}_{1}$), it resumes transmitting data. This should not be used with $^{Q}_{1}/^{S}_{3}$ handshaking. Therefore, if XmitPace is enabled, InhHndShk (G) and Inh$^{Q}_{1}$ (H) in terminal configuration must both be set to "YES", which disables $^{Q}_{1}/^{S}_{3}$ handshaking. If this field is set to "NONE", the terminal does NOT recognize the ASCII $^{Q}_{1}$ and $^{S}_{3}$ codes as XON and XOFF. For another form of transmit pacing, refer to the description of the CS(CB)Xmit field above. Values: NONE XON/XOFF

[f8] config keys

Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following:

(this label is blank on the 2622A)

POINT-TO-POINT PROGRAMMING INFORMATION

This topic discusses programming information of interest to someone who is writing a data communications driver or controller program to communicate with the terminal in an asynchronous point-to-point environment.

An asynchronous point-to-point data communications environment is characterized by a flow of characters that

have been produced over random time intervals. In order to achieve hardware synchronization, each character is delimited by a "start bit" and one or more "stop bits".

Start And Stop Bits

These hardware-generated bits are used for synchronizing the transmit and receive devices in an asynchronous environment. A start bit is a "zero" line state (+12V) that lasts for 1.0 bit time; it is affixed to the beginning of a serial character bit stream (which may also include a parity bit). A stop bit is a mark or a "one" line state (−12V) that lasts for 1.0 bit time; it is appended to the end of each serial character bit stream. After the stop bit, the line remains in the mark state until the next character, signified by a start bit, is transmitted.

The start and stop bits are not configurable. For 110 baud, the terminal is automatically configured to transmit 2 stop bits with the data to the computer. At 110 baud, the terminal also expects to receive 2 stop bits with the data received from the computer. For all other baud rates, 1 stop bit is transmitted with data and 1 stop bit is expected to be received with data.

Parity Checking

In an asynchronous point-to-point environment, the terminal provides a vertical redundancy check (VRC), which is a character-based error checking mechanism for non-binary data. With VRC, an additional bit is affixed to each character to provide an expected high-order bit state for each character. This type of parity generation and checking is a means of determining the validity of data transfer on a character-by-character basis.

Note that when 8-bit data is being exchanged, parity cannot be used and the "Parity" field in the datacomm configuration menu must be set to "NONE". Otherwise, one of the bits will be mistaken for parity.

The terminal offers the following four types of parity:

1. 0'S. The high-order bit is always a zero.
2. 1'S. The high-order bit is always a one.
3. ODD. The high-order bit is set to a zero or a one, whichever produces an odd number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).
4. EVEN. The high-order bit is set to a zero or a one, whichever produces an even number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).
5. NONE. Eight bits of data are transmitted and received. No parity bit is transmitted or received.

Receive Buffer

The terminal's receive buffer is a first in/first out (FIFO) storage area for accepting data from the remote device. When you are using any type of receive pacing, this buffer is partitioned into a working buffer and a 40-byte overrun area. For example, the specified buffer size is always 256 bytes, thus if receive pacing is being used, the working buffer is 216 bytes long and the overrun area is 40 bytes long. When the data being received exceeds the working buffer and intrudes on the overrun area, the terminal will exercise its receive pacing mechanism (send an XOFF, for example, if XON/XOFF receive pacing is enabled) at that time to temporarily halt the flow of data from the remote device. When enough data has been processed so that the receive buffer is only half full, the terminal then signals the remote device to resume transmission (by sending an XON, for example, if XON/XOFF receive pacing is enabled).

There is no equivalent overrun area for transmitting data from the terminal to the remote device.

Receive Errors

When receiving data from the remote device, the terminal can detect the following three types of error conditions (in addition to parity errors):

1. Character overruns—a character is received before the preceding character was processed by the terminal's datacomm firmware.
2. Framing errors—no stop bit was detected at the end of a character.
3. Buffer overflows—the entire allocated buffer space is filled (both the working buffer and the overrun area).

Receive errors, when detected, are reported to the remote device by way of byte 5 of the primary terminal status bytes. The remote device will not be able to determine which type of error occurred. If multiple receive errors occur simultaneously, only one will be reported.

When a datacomm receive error occurs, a delete character (■) is placed in the datacomm queue and later it is displayed on the terminal screen.

NOTE

Because null and del characters are automatically stripped from datacomm, the only del characters appearing on the screen from datacomm are the result of datacomm errors.

Local/Remote Modes

The data communications portion of the terminal operates independently whether the terminal is in local or remote mode. If the terminal is switched from remote to local while

data is being received from the remote device, the datacomm portion of the terminal continues receiving data (it does NOT halt the transmission). In such a case, the data received while the terminal is in local is discarded by the terminal's maincode firmware.

Full-Duplex Operation

In a full-duplex environment, the terminal is capable of transmitting and receiving data simultaneously. The ability to transmit may be inhibited temporarily, but it is never exclusive of the ability to receive. Two physical sets of data lines are required; control lines are needed only when hardware handshaking or a modem is used. Transitions on the control lines have no effect on the actual transmit/receive state of the terminal.

When the terminal is connected to the host computer via a modem, the following primary control lines are required:

Request to Send (RS/CA)

Clear to Send (CS/CB)

Data Terminal Ready (CD/TR)

If the terminal is hardwired directly to an HP 3000 computer system (no modem), only Transmit Data (SD/BA), Receive Data (RD/BB), and Signal Ground (AB) are required.

Pacing Mechanisms

In a full-duplex environment, the terminal can participate in either of the following forms of transmit pacing:

1. Hardware handshake. The host computer can temporarily restrain the terminal from transmitting by lowering the Clear to Send (CB) line. Note that this type of transmit pacing can only be used in a hardwired configuration where the Clear to Send (CB) line exists in the cabling.
2. XON-XOFF handshake. The host computer or external printer uses the ASCII control codes XON (P_1) and XOFF (P_3) to start and stop the terminal from transmitting. Note that a single XON code cancels any number of preceding XOFF codes.

In a full-duplex environment, the terminal can also participate in the XON/XOFF handshake form of receive pacing, in which the terminal uses the ASCII control codes XON (P_1) and XOFF (P_3) to start and stop the host computer from transmitting. Note that a single XON code cancels any number of XOFF codes.

The terminal can also participate in an ENQ/ACK handshake (which is a Hewlett-Packard handshaking mechanism). With this form of handshaking, the host computer transmits a block of data and then sends an ASCII <ENQ> control code. The terminal responds to the <ENQ> by sending back an ASCII <ACK> control code when it has processed all of the data preceding the <ENQ>. The general interpretation of these two control codes is as follows:

ENQ: "Have you processed the data up to this point?"
ACK: "Yes, I have."

The above pacing mechanisms are responded to by the terminal in the following order of precedence:

1. Hardware handshaking pacing (highest priority)
2. XON/XOFF transmit pacing
3. XON/XOFF receive pacing
4. ENQ/ACK pacing (lowest priority)

Status

SECTION VII

INTRODUCTION

This section tells how a program executing in a host computer obtains and interprets status information from the terminal.

Status requests are issued in the form of escape sequences. There are five types of status requests:

1. Terminal ID Status. This request is the means by which your program verifies what kind of terminal it is communicating with.
2. Primary Terminal Status. This request returns seven bytes that report the status of some of the latching keys, various error and pending flags, and the following configuration menu fields (see Section II, table 2-1 for descriptions of the fields):

   Xmit Fnctn
   SPOW
   Inh Eol Wrp
   Line/Page
   Inh Hndshk
   Inh DC2

   Esc Xfer
   Compat
3. Secondary Terminal Status. This request returns seven bytes that report the status of the memory lock, buffer memory, and I/O firmware.
4. Device Status. This request returns three bytes that report the status of the integral printer (and the external device on the HP 2623A).
5. Graphics Status. These requests apply to the HP 2623A Graphics Terminal. There are 12 graphics status requests that can be made; they are listed in table 7-2.

The escape sequence used for each of the above requests and the format of the returned status information is presented in the following paragraphs.

All status requests are treated as block transfers. In response to a status request, the terminal transmits an escape sequence, followed by a series of bytes, followed by a terminator. The terminator is as follows:

Character Mode:	C_R or $C_R L_F$
Block Line Mode:	C_R or $C_R L_F$
Block Page Mode:	<BlkTermnator>

In either character mode or block line mode, the $C_R L_F$ is used if auto line feed mode is enabled. In block page mode, the block terminator is <Blk Termnator> (E_S).

The type of handshaking used is determined by the setting of the InhHndShk and Inh D_2 fields of the configuration menu as follows:

1.	InhHndShk(G)	= YES	No handshake
	Inh DC2(H)	= YES	
2.	InhHndShk(G)	= NO	D_1
	Inh DC2(H)	= YES or NO	
3.	InhHndShk(G)	= YES	$D_1/D_2/D_1$
	Inh DC2(H)	= NO	

INTERPRETING STATUS

For primary, secondary, and device status requests, the terminal returns an escape sequence followed by a string of bytes. The status information is contained in the lower four bits of each byte. The upper four bits are set so that the byte translates into one of the 16 ASCII characters shown in table 7-1.

For a terminal ID request, the terminal returns the 5-character ASCII string "2622A" or "2623A".

Terminal ID Status

You request the terminal ID status by issuing the following escape sequence:

$E_C * s \wedge$

The terminal responds by sending back the following five-character string:

2622A
or
2623A

Table 7-1. ASCII Status Characters

ASCII CHARACTER	BINARY
0	0011 0000
1	0011 0001
2	0011 0010
3	0011 0011
4	0011 0100
5	0011 0101
6	0011 0110
7	0011 0111
8	0011 1000
9	0011 1001
:	0011 1010
;	0011 1011
<	0011 1100
=	0011 1101
>	0011 1110
?	0011 1111

Terminal Status

Terminal status is made up of 14 status bytes (bytes 0–13) containing information such as display memory size, switch settings, configuration menu settings, and terminal errors. These 14 status bytes are displayed below the self-test screen pattern when the "TERMINAL TEST" (f5) key (in the "service keys" set of function keys) is pressed. There are two terminal status requests: primary and secondary. Each returns a set of 7 status bytes.

PRIMARY TERMINAL STATUS. You request the first set of terminal status bytes (bytes 0–6) by issuing the following escape sequence:

ESC ^

The terminal responds with an ESC \, and seven status bytes followed by a terminator. A typical primary terminal status request and response is illustrated in figure 7-1. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.

SECONDARY TERMINAL STATUS. You request the second set of terminal status bytes (bytes 7–13) by issuing the following escape sequence:

ESC ~

The terminal responds with an ESC |, and seven status bytes followed by a terminator. A typical secondary terminal status request and response is illustrated in figure 7-2. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.

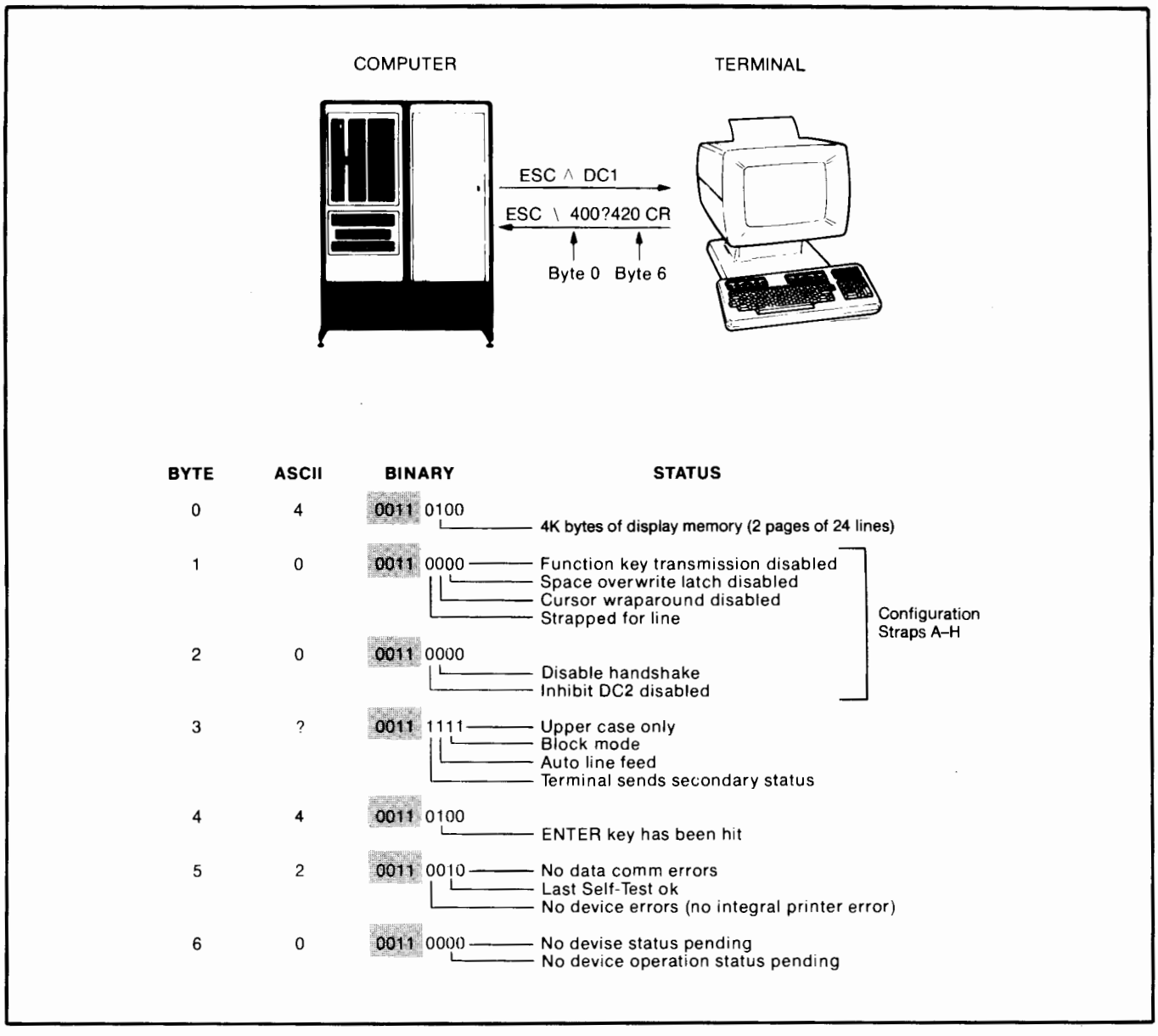
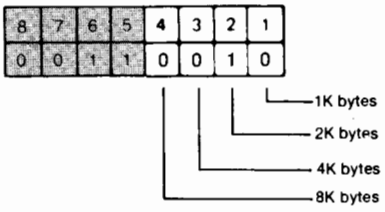


Figure 7-1. Primary Terminal Status Example

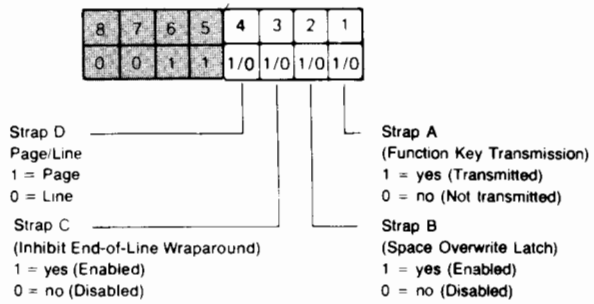
PRIMARY STATUS BYTES

BYTE 0 DISPLAY MEMORY SIZE



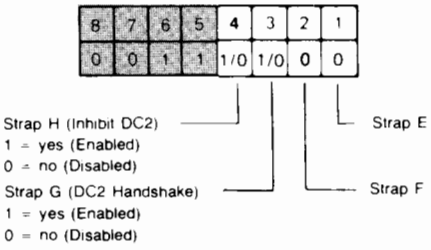
This byte specifies the amount of display memory available in the terminal. Note that the HP 2622A and HP 2623A terminals always return 4K bytes.

BYTE 1 CONFIGURATION STRAPS A–D



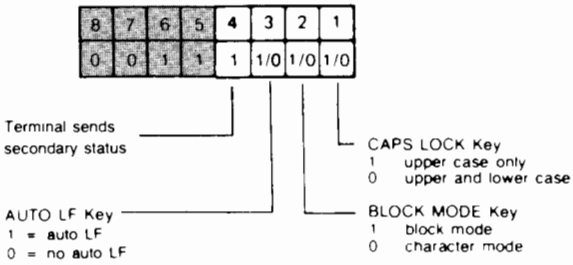
Refer to Section II for a detailed description of configuration straps A–D.

BYTE 2 CONFIGURATION STRAPS E–H

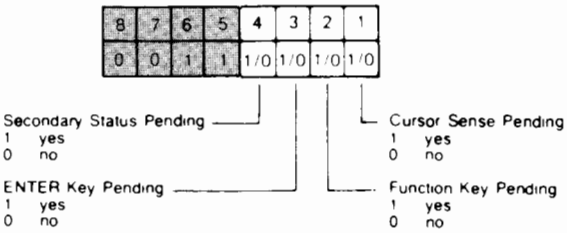


Refer to Section II for a detailed description of configuration straps G and H (straps E and F do not apply to the terminal).

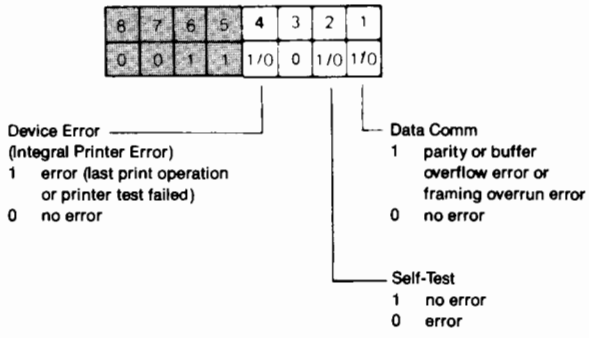
BYTE 3 LATCHING KEYS



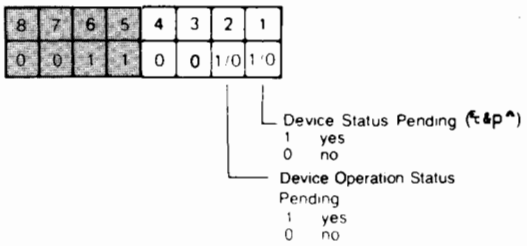
BYTE 4 TRANSFER PENDING FLAGS



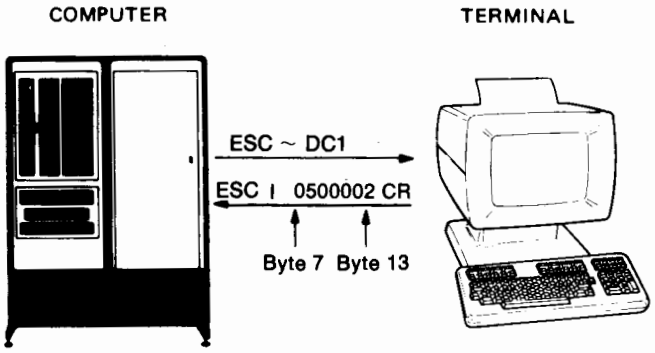
BYTE 5 ERROR FLAGS



BYTE 6 DEVICE TRANSFER PENDING FLAGS



(tracks "S", "F", or "U" completion codes associated with ESC&p device control sequences.)



BYTE	ASCII	BINARY	STATUS
7	0	0011 0000	
8	5	0011 0100	I/O firmware installed Terminal identifies self
9	0	0011 0000	
10	0	0011 0000	Do not send escape codes to printer Compatibility Mode off (2623A)
11	0	0011 0000	
12	0	0011 0000	
13	2	0011 0010	Memory locked

Figure 7-2. Secondary Terminal Status Example

SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY
(always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- 1K bytes
- 2K bytes
- 4K bytes
- 8K bytes

Memory installed in addition to display memory that is available for use as data buffers. Note that the HP 2622A and HP 2623A terminals always return a 0 value.

BYTE 8 TERMINAL FIRMWARE CONFIGURATION

8	7	6	5	4	3	2	1
0	0	1	1	0	1	0	1

- 0 = Non-programmable terminal
- 1 = Terminal identifies self
- 1 = I/O firmware installed, integral printer (Option 50) present
- 0 = No APL Firmware

APL Firmware does not apply. Note that the HP 2622A returns either a 4 or a 5; the HP 2623A always returns a 5.

BYTE 9 CONFIGURATION STRAPS J–M
(always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- Strap M
- Strap J (Auto Terminate): 1 = yes (Enabled), 0 = no (Disabled)
- Strap K (Clear Terminator): 1 = yes (Enabled), 0 = no (Disabled)
- Strap L (Self-Test Inhibit): 1 = yes (Inhibit test), 0 = no (Allow test)

Straps J–M do not apply to the terminal.

BYTE 10 KEYBOARD INTERFACE KEYS (N–R)

8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	1/0	1/0

Switch R

Q	P	
0	0	Compatibility Mode off
0	1	Scaled Compatibility mode
1	0	Unscaled Compatibility mode
1	1	Compatibility Mode off

- Switch N Printer (Escape Code Transfer): 1 = open (Send ESC code), 0 = closed (Do not send code)
- Switch P Compatibility Mode (Scaled): 1 = Enabled, 0 = Disabled
- Switch Q Compatibility Mode (Unscaled): 1 = Enabled, 0 = Disabled

Straps N–R do not apply to the HP 2622A terminal.

BYTE 11 CONFIGURATION STRAPS S–V
(always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- Strap V
- Strap S
- Strap U
- Strap T

Straps S–V do not apply to the terminal.

BYTE 12 CONFIGURATION STRAPS W–Z
(always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

- Strap Z
- Strap W (Data Comm Test): 1 = yes (Inhibit), 0 = no (Allow)
- Strap Y
- Strap X

Straps W, X, Y, and Z do not apply to the terminal.

BYTE 13 MEMORY LOCK MODE

8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	1/0	1/0

- 1 = locked in row ≠ 0; 0 = locked in row 0 (memory overflow protect)
- 1 = memory lock on; 0 = memory lock off
- 1 = memory full, 0 = memory not full } only active when memory lock is enabled

DEVICE STATUS

The following information on device status only applies when option 050 (the integral printer) is installed and the external device port on the HP 2623A.

The status of the integral printer and external printer can be obtained by issuing a device status request. This request would typically be made following a print operation or after examining bytes 5 and 6 of the primary status. The device status bytes are shown in figure 7-3.

You request device status by issuing the following escape sequence:

`Esc&p <device code>^`

where `<device code>` is either 4 or 6. Both 4 and 6 are interpreted as the integral printer on the 2622A. The external printer is 4 on the 2623A. However, you may obtain status of the integral printer by device code 4 if "`INT`" (integral) is configured in the "`Printer Code 4`" field of the External Drive Configuration Menu of the HP 2623A.

If `<device code>` is any value other than 4 or 6, the escape sequence is ignored.

The terminal responds with the sequence `Esc\p <device code>`, followed by three status bytes followed by a terminator. A typical device status request and response are illustrated in figure 7-4.

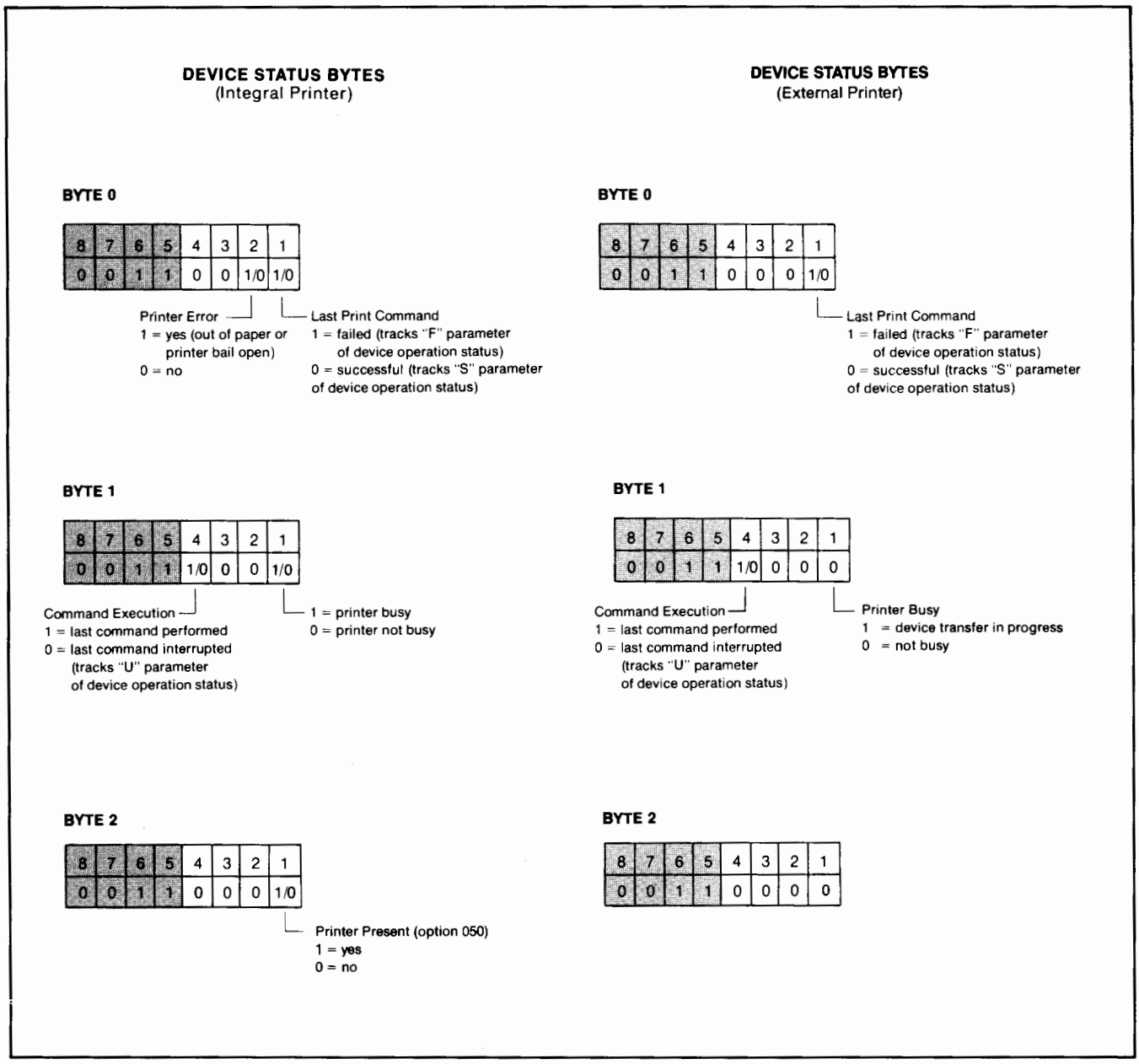


Figure 7-3. Device Status Bytes

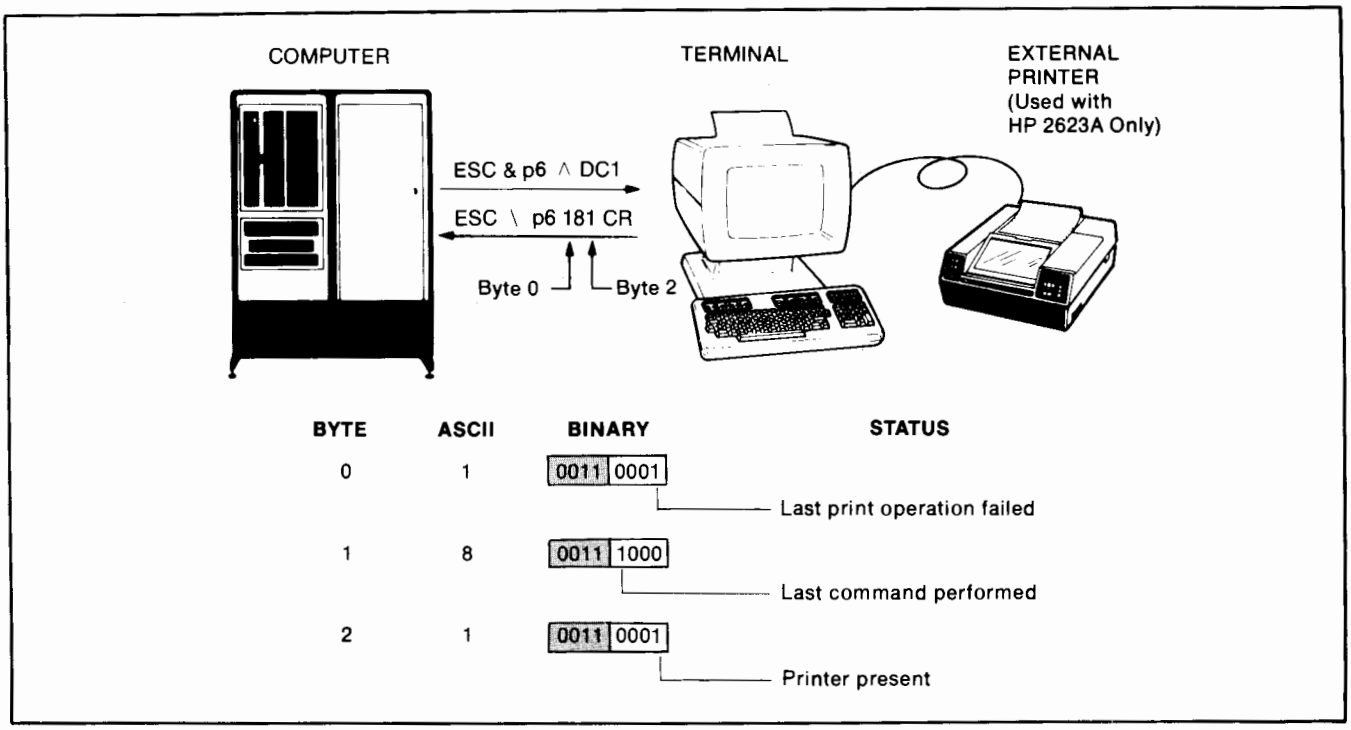


Figure 7-4. Device Status Example

HP 2623A GRAPHICS STATUS

In addition to normal HP 2623A terminal status you can request graphics status information. All graphics status requests are initiated by sending an Et *s followed by a single parameter (1 through 12) followed by a ^. The single parameter selects the particular status block desired. If an invalid parameter is used, the terminal will simply return its I.D. (see Device I.D. Request, parameter = 1).

Graphics Status Request: Et * s <parameter> ^

where: Et * s is the graphics status escape sequence.

<parameter> is 1-12 and selects one of twelve blocks of graphics status information.

The graphics status blocks that can be requested are listed in table 7-2 together with the format of ther terminal's response. Detailed descriptions of each of the status requests are contained in the following paragraphs.

The terminal will respond with one or more bytes of status information followed by a block terminator. All status information is returned in ASCII format, separated by commas. Coordinates are returned in a fixed format, consisting of a sign and five digits. Leading zeros are used as required to provide a fixed number of digits (i.e. +00100, −01234). This allows you to use simple input statements without the need to mask or shift bits.

If the DC1 handshake protocol is enabled (i.e., "NO" is entered in the "InhHndShk(G)" and "Inh DC2(H)" fields of the Terminal Configuration menu—refer to Section II), the status block is not actually sent until receipt of a DC1 character. If the DC1 character is used, only one status request can be enabled while the terminal is waiting for a DC1. When the DC1 is received, the last status block requested will be sent.

While the terminal is waiting for the DC1, the Device Status Pending bit is set (see byte 6 of the primary status bytes on page 7-3).

The terminal's configuration determines the terminating character sent following the status block (CR, CR-LF, or RS)—refer to page 7-1. Graphics status requests turn on an echo suppress mode in the terminal. This prevents information echoed back from the computer from being displayed on the screen. Once a status block has been sent, characters received by the terminal will not be displayed until one of the following control characters is received: Q, RS, CR, Et, GS, H, LF, RS, U, Y. With the exception of CR and LF the terminating control code itself will be executed.

The terminal expects the status information to be echoed and uses the terminating control character to turn off the suppress echo mode. If the computer does not echo the status back, a suitable control character must be returned to the terminal to turn off the echo suppress mode.

The graphics status blocks that can be requested are shown in table 7-2.

Table 7-2. HP 2623A Graphics Status Requests

Parameter	Request	Response
1	Read device I.D.	2623A
2	Read current pen position	<X>,<Y>,<PEN>
3	Read graphics cursor position	<X>,<Y>
4	Read graphics cursor position with wait	<X>,<Y>,<KEY>
5	Read display size	<LLX>,<LLY>,<URX>,<URY>,<MMX>,<MMY>
6	Read device capabilities	<b1>,<b2>,. . .<b15>,<b15>
7	Read graphics text status	<X size>,<Y size>,<origin>,<angle>,<slant>
8	Read zoom status	001.,0
9	Read relocatable origin	<X>,<Y>
10	Read Reset status	<RESET>,<b1>. . .<b6>,<b7>
11	Read area shading capability	1,8,8
12	Read dynamics capability	1,1

Read Device I.D. (Parameter=1)

When you request a device I.D. the terminal responds with its Hewlett-Packard model number, 2623A.

Device I.D. Request: `Esc * s 1 ^`

The terminal responds: 2623A <terminator>

Read Current Pen Position (Parameter=2)

The pen position and status are returned as a string of ASCII characters.

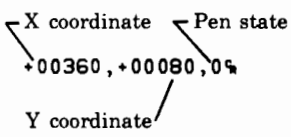
Pen Position Request: `Esc * s 2 ^`

The terminal responds: <X>,<Y>,<Pen>,<terminator>

where: <X> = X coordinate
<Y> = Y coordinate
<Pen> = Pen state, 0=pen up, 1=pen down

For example, assume that the pen is at 360, 80, the pen is up, and the terminal is set for the DC1 handshake, with CR as the terminator:

The computer sends: `Esc * s 2 ^` <terminator> DC1



Read Graphics Cursor Position (Parameter=3)

The graphics cursor position is returned as a string of ASCII characters.

Read Graphics Cursor Request: `Esc * s 3 ^`

The terminal responds: <X>,<Y> <terminator>

where: <X> = X coordinate
<Y> = Y coordinate

Read Cursor Position with Wait (Parameter=4)

This request allows the user to position the cursor, then strike a key to return the position. The ASCII decimal code for the key struck is also returned (not the actual character). The code is returned as three digits. For example, striking an uppercase A would return 065. Only ASCII character keys will generate a response (i.e. ROLL UP, ROLL DOWN, etc. are ignored). The graphics cursor is turned on, if not already on. If an escape sequence is received by the terminal after it has received the READ CURSOR with WAIT command and before a key is struck, the READ CURSOR command will be aborted. The new sequence will be executed instead.

Read Graphics Cursor with Wait Request: `Esc * s 4 ^`

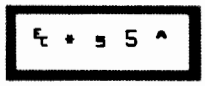
The terminal responds: <X>,<Y>,<key code>
<terminator>

where: <X> = X coordinate
<Y> = Y coordinate
<key code> = Decimal value of key struck

The position bytes are ordered as in the read pen request. The decimal of ASCII characters are given in ASCII character set tables in Appendix B.

Read Display Size (Parameter=5)

This request returns the number of displayable units in the X and Y axes. It also returns the number of units per millimeter in the display. This request allows you to scale data for use on graphic devices with varying display areas.

Read Display Size Request: 

The terminal responds: <LLX>,<LLY>,<URX>,<URY>,
<MMX>,<MMY><terminator>

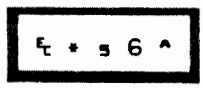
where: <LLX>,<URX> = Lower left and upper right x coordinates
<LLY>,<URY> = Lower left and upper right y coordinates
<MMX>,<MMY> = number of units per millimeter in the x and y axes, (five digits and a decimal point)

The terminal will always return a fixed response. The lower left corner has coordinates of 0,0. The upper right corner has coordinates of 511,389. There are approximately 2 units per millimeter in each axis.

Terminal response: +00000,+00000,+00511,+00389,
00002.,00002.<terminator>

Read Device Capabilities (Parameter=6)

The device capabilities request returns a list of graphic and plotting features available in the terminal. This allows you to use one program for a variety of graphic devices. Not all of the features listed are available in the terminal. The absence of a feature is indicated by a 0. If a feature is present, it may be necessary to send an additional request to determine the exact capabilities present. Where multiple response values are possible the terminal's standard response is shaded.

Device Capability Request: 

The terminal responds:
<b1>,<b2>,<b3>,<b4>,<b5>,<b6>,<b7>,
<b8>,<b9>,<b10>,<b11>,<b12>,<b13>,<b14>,
<b15>,<b16><terminator>

where: <b1> = Clear Display
0 = no clear
1 = paper advance
2 = clear (total erase)

<b2> = Number of Pens (1)

<b3>,<b4> = Not Used (0,0)

<b5> = Area Shading
0 = no

<b6>,<b7> = Not Used (0,0)

<b8> = Dynamic Modification
0 = no

<b9> = Graphics Character Size
0 = fixed
2 = any size

<b10> = Graphics Character Angles
0 = fixed
2 = multiples of 45°
3 = any angle

<b11> = Graphics Character Slant
0 = fixed
2 = any angle

<b12> = Dot-Dash Line Patterns
0 = none
1 = predefined only

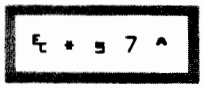
<b13>-<b16> = Not Used (0,0,0,0)

The terminal will always respond:

3,1,0,0,1,0,0,1,1,1,1,2,0,0,0,0<terminator>

Read Graphics Text Status (Parameter=7)

The terminal returns the current text size, orientation, slant, and type of justification. Refer to Section X, Graphics Control Functions for a description of graphics text characteristics.

Read Graphics Text Request: 

The terminal returns: <x size>,<Y size>,<origin>,
<angle>,<slant><terminator>

where: <X size> = X dimension of the character cell (sign + 5 digits)
<Y size> = Y dimension of the character cell (sign + 5 digits)

<origin> = Relative position of text to cursor (see text origin command)(1 digit)
<angle> = Text angle 0, 90, 180, or 270 (five digits and a decimal point)
<slant> = 00000. or 00045. degrees

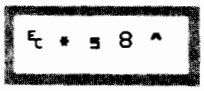
Example terminal response:

+00007,+00010,1 <terminator>

Read Zoom Status (Parameter=8)

This request returns the zoom setting. Since the HP 2623A terminal does not have the zoom feature, it always returns constant values.

Read Zoom Status Request:



The terminal responds:

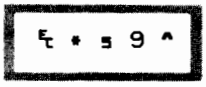
<zoom size>,<zoom on/off><terminator>

where: <zoom size> = 001.
<zoom on/off> = 0 for Off

Read Relocatable Origin (Parameter=9)

The position of the relocatable origin is returned as x and y coordinates.

Read Relocatable Origin Request:



The terminal responds:

<X coordinate>,<Y coordinate><terminator>

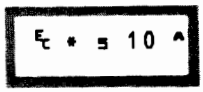
Example terminal response:

+00000,+00000 <terminator>

Read Reset Status (Parameter=10)

You can determine whether or not the terminal has executed a full reset (or Power On) since the last time reset status was checked. This will tell you whether or not you need to reestablish terminal settings or images before resuming terminal functions. An additional seven bytes are returned but are not used.

Read Reset Status Request:



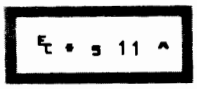
The terminal responds: <reset>,<b1>,<b2>,<b3>,<b4>,<b5>,<b6>,<b7><terminator>

where: <reset status> = 0 No full reset since last check
or
1 Terminal has been reset
<b1>-<b7> = 0 (not used)

Read Area Shading Capability (Parameter=11)

The area shading capability of the terminal can be read. These are fixed for the terminal.

Read Area Shading Request:



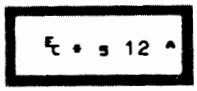
The terminal will always respond: 1,8,8 <terminator>

The "1" indicates that the area shaded must be rectangular. The first "8" indicates that the shading pattern is 8 units wide. The second "8" indicates that the shading pattern is 8 units high.

Read Graphic Modification Capabilities (Parameter=12)

You can read the terminal's dynamic graphics capabilities. This is the ability of the terminal to change selected portions of the display. These are fixed for the terminal.

Read Graphic Modification Capabilities Request:



The terminal will always respond: 1,1 <terminator>

These two bytes indicate that the terminal has selective erase and compliment capabilities.

Any Other Parameter

Any other parameter which has not been assigned causes the terminal I.D. to be returned. This is to prevent an invalid status request from tying up the requesting computer while waiting for a response.

2623A <terminator>

Error Messages and Self-Test

SECTION VIII

INTRODUCTION

This section is divided into two portions. The first discusses the various error messages that may appear on the terminal's screen while you are attempting to perform operations through the keyboard. The second discusses the various types of self-tests that are incorporated into the terminal.

ERROR MESSAGES

When the terminal detects a parameter inconsistency or error condition, it locks the keyboard and displays an appropriate error message across the bottom of the screen (replacing the function key labels). Press RETURN to unlock the keyboard, clear the message, and reinstate the current function key labels.

The various possible error messages and their general meanings are as follow:

`Default configs used`
`Press RETURN to clear`

This message is displayed when the terminal attempts to read the content of non-volatile memory but detects a CRC error (e.g., at power-on time, during a hard reset).

To determine whether the problem is a bad battery or a bad RAM chip, run the Terminal Test described later in this section. If the RAM chip used for non-volatile memory is bad, the Terminal Test will fail and generate an appropriate "`CMOS RAM`" message identifying the faulty chip. If the test passes, then the "`default configs used`" power-on message indicates that the battery needs to be changed. Instructions on how to change the battery are provided in Section IX, "Terminal Maintenance Procedures".

After clearing the message (by pressing RETURN), you may then reconfigure the terminal as you desire.

`Integral printer error`
`Press RETURN to clear`

Something is wrong with the integral printer. It may just be out of paper or the metal latch (under the plastic printer lid) may not be pressed down securely.

`No 'TO' device`
`Press RETURN to clear`

You attempted to initiate a device control data transfer (copy line, copy page, copy all) but no destination device is currently defined. Press RETURN, use the "`device control`" set of function keys to define an external printer and/or the integral printer as the "to" device, and then retry the copy operation.

TERMINAL SELF-TESTS

The terminal includes six types of self-tests:

- Power-On Test
- Manufacturing Test
- Terminal Test
- Identify ROMs
- Datacomm Test
- Internal Printer Test

The Power-On Test is automatically initiated as the result of a power-on sequence. All of the other tests must be initiated using the "`service keys`" (except the Terminal Test, which can also be initiated programmatically or by using a "`MODES`" function key).

Power-On Test

The Power-On Test, which is performed automatically whenever you turn on the terminal's power, does the following:

1. Tests the processor and verifies the integrity of all ROM (Read-Only Memory) and RAM (Random-Access Memory) chips within the terminal.

If the Power-On Test results are normal:

1. at power on, the terminal beeps once,
2. does the test for about 15 seconds,
3. beeps once again,
4. brings up the `MODES` group of softkey labels on the terminal screen.

If an error is found, one of the following will occur when the terminal is turned on:

a. The terminal will fail to beep at all.
b. The terminal will beep continuously.
c. After the first beep, the terminal will beep 1 to 14 times and no softkey labels will appear on the screen.

If the terminal fails to beep, check to make sure the keyboard is connected properly, and try again. If the problem persists, call the nearest HP Sales and Service Office and arrange to have the terminal repaired.

If one of the other error conditions occur (beeping too many times), also call the nearest HP Sales and Service Office and arrange to have the terminal repaired.

Terminal Test

This test does the following:

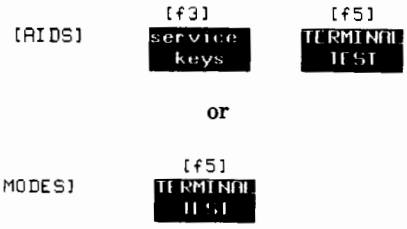
1. Performs a graphics test on the 2623A. The test checks both the vector generating function as well as the graphics memory. If this test fails, a message indicating the failing component may be displayed. The display shows two patterns:

The first:

The second:

2. Displays the message "TESTING" at the bottom of the screen (on the same line where the function key labels normally appear).
3. Verifies the integrity of all firmware ROM chips within the terminal.
4. Non-destructively verifies the integrity of all RAM chips within the terminal (including the one used for non-volatile memory).
5. Displays the test pattern shown in figure 8-1 or 8-2 (depending on whether you have the optional Line Drawing or National character sets).

To initiate the Terminal Test press the following keys in the sequence shown:



If a ROM error is detected, the following message is displayed across the bottom of the screen:

```
ROM ERR #x
Press RETURN to clear
```

where "x" will be a number from 1 to 6. This message contains information identifying the bad ROM chip(s) and describing the nature of the detected error condition. In such a case, or for any other error message, write down the message so you can relate it accurately to your HP Service Representative over the telephone (this allows him to arrive prepared with the proper replacement parts).

If a RAM error is detected, the following message is displayed across the bottom of the screen:

```
RAM ERR #x
Press RETURN to clear
```

where "x" is 1 to 8. This message also contains information identifying the bad RAM chip(s) and describing the nature of the detected error condition. Write down the message and call your nearest HP Service Representative.

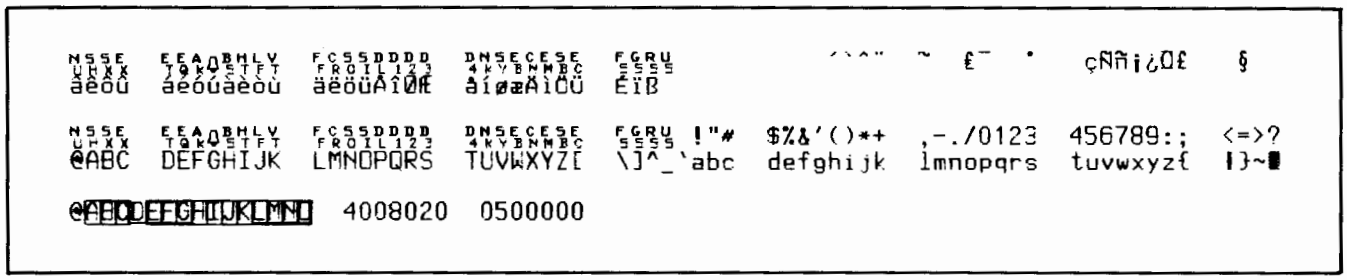


Figure 8-1. Screen Test Pattern, Standard Terminal

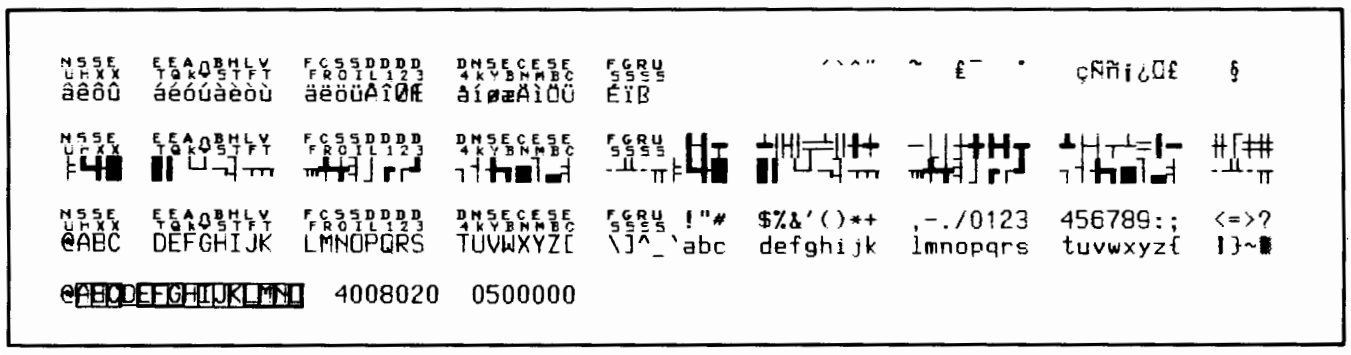


Figure 8-2. Screen Test Pattern, Option 202 (Line Drawing Character Set)

If the ROM and RAM chips all pass the test but the test pattern on the screen is malformed, then this would suggest a problem with the video portion of the terminal (the sweep mechanism, the yoke alignment, and so forth).

Identify ROMS

To generate a descriptive list of all ROM chips installed in the terminal, press the following keys in the order shown:

[AIDS] [f3] service keys [f6] IDENTIFY ROMS

A list similar to the one shown in figure 8-3 is displayed on the screen.

Datacomm Test

The data communications (datacomm) self-test checks the 50-pin port at the rear of the terminal. It also checks the 25-pin external peripheral port of the HP 2623A.

To enable the data communications (datacomm) self-test press the following keys in the sequence shown:

[AIDS] [f3] service keys [f7] DATACOMM TEST

Test hood (HP part no. 02620-60056) must be connected to the 50-pin datacomm port, and test hood (part no. 02620-60062) must be connected to the 25-pin external peripheral port (HP 2623A only) to perform the datacomm test. Otherwise, "Datacomm Error 1" will appear on the screen if the test is attempted without them being connected.

Data Communications Test Hoods

Part No.	Description
02620-60056	This is a male 50-pin test hood for use on the datacomm port on both the HP 2622A and HP 2623A terminals.
02620-60062	This is a male RS-232C test hood for use on the external peripheral port on the HP 2623A terminal.

HP 2622A	HP 2623A
Firmware ROMs	Firmware ROMs
1818-1685 2123	1818-1640 2129
1818-1686 2123	1818-1642 2129
1818-1687 2123	1818-1455 2129
1818-1688 2123	1818-1454 2129
	1818-1641 2129
	1818-1781 2129

Figure 8-3. ROM Identification Listing

These test hoods are available as a set which can be ordered separately as the HP 13259A Data Comm Self-Test Connector Kit.

The loopback test consists of a data loopback operation (which checks to see if the character sent is the same as the character received), a baud rate test (which verifies that the baud rate mechanism is functioning properly within ±2% of the configured baud rate), and a modem control line test.

While the test is executing, "Testing" is displayed on the terminal screen. If no errors are found, the terminal beeps and displays the softkeys. If an error is found, an error message will appear on the terminal screen, similar to a ROM error.

The "Err#" field contains a numeric error code which is interpreted as follows:

1 = Test connector not present
2 = Baud rate too fast
3 = Baud rate too slow
4 = Error in Control lines
5 = Character did not loop back
6 = Received character NOT same as one transmitted
7 = Framing error in character
8 = A character was overrun

Printer Test

The Printer Test, which is usable only if the optional integral printer is present, exercises all the features of the integral printer to verify that it is functioning properly. To initiate this test press the following keys in the sequence shown:

[AIDS] [f3] service keys [f8] INT PRT TEST

If the printer is functioning properly it generates the test pattern shown in figure 8-4.

Note that if your terminal does NOT include the integral printer the f8 key label in the "service keys" set of function keys will be blank and pressing that key will have no effect.

If an error condition is detected while the test is being executed, the message

INTEGRAL PRINTER ERROR
Press RETURN to clear

appears across the bottom of the screen. To clear the message, press RETURN. Note that the error condition may be

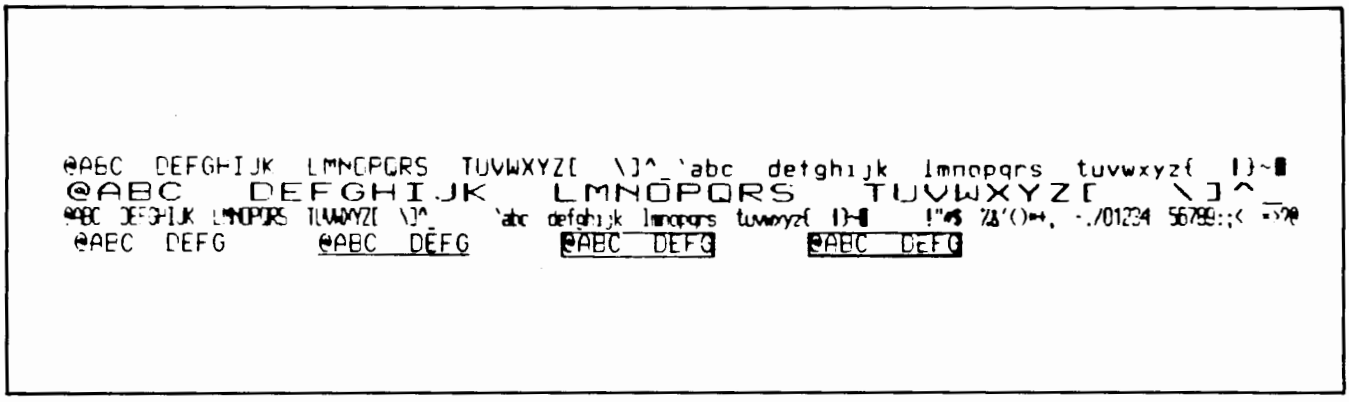


Figure 8-4. Integral Printer Self-Test Output

either of the following, in which case you could correct it yourself:

1. Out of paper.
2. The metal latch (under the plastic printer lid) is not pressed down securely.

If the error is not due to one of these conditions, call your nearest HP Service Representative.

Terminal Maintenance Procedures

SECTION IX

INTRODUCTION

This section provides information on preventive maintenance for your terminal, such as cleaning the screen and keyboard; and routine maintenance such as replacing paper in the integral printer.

CLEANING THE SCREEN AND KEYBOARD

The display screen and the keyboard should be cleaned regularly to remove dust and grease. First, lightly dust the entire terminal using a damp, lint-free cloth or paper towel. The cloth or paper towel should be damp enough to pick up any dust, but should not be wet. Avoid wiping dust or lint into the key area of the keyboard.

Greasy smudges and fingerprints can be removed using most conventional spray cleaners. Avoid spraying between the keys.

DO NOT use petroleum-based cleaners (such as lighter fluid) or cleaners containing benzene, trichlorethylene, ammonia, dilute ammonia, or acetone because these chemicals could damage the terminal's plastic surfaces.

BATTERY MAINTENANCE

The non-volatile portion of memory that contains the terminal's configuration data is protected against destruction by a battery that is located just above the rear panel of your terminal. Figure 9-1 shows the rear panel and the location of the battery.

The battery requires no special care or maintenance. It should, however, be replaced with a new battery every 12 months. You may purchase a replacement battery through conventional retail stores. When doing so, request a Mallory Battery, Type TR133. You may also order replacement batteries through your local HP Sales and Service Office using the following nomenclature and part number:

HP 2622A or HP 2623A Battery,
HP Part No. 1420-0259

If your terminal includes the optional thermal printer you may wish to record the various configuration menus on paper before removing the old battery. To do so, perform the following:

1. Use the "`config keys`" set of function keys to display the particular menu of the screen.
2. Press the ENTER key.

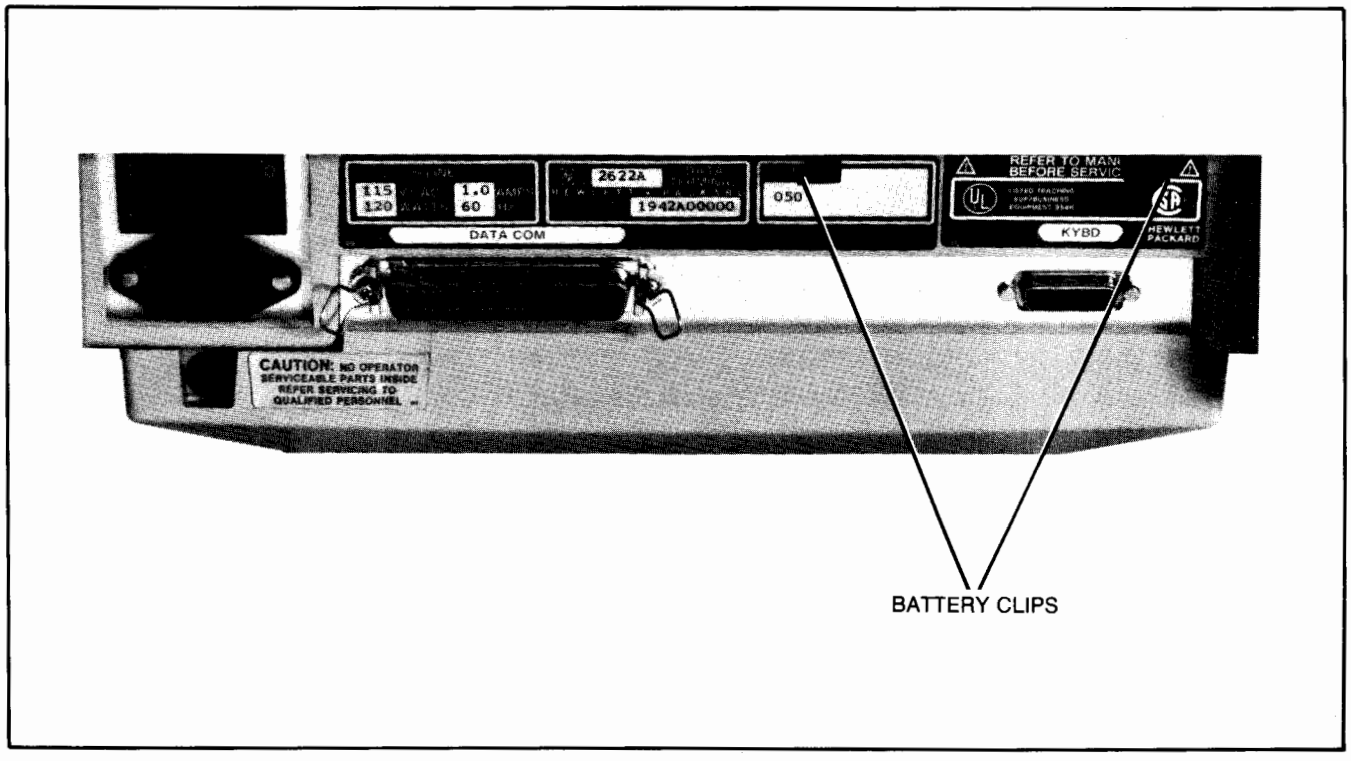


Figure 9-1. Battery Location, Rear Panel

Replace the battery as follows:

1. Grasp the battery support at points A and B as shown in figure 9-2.
2. Squeeze the tabs at points A and B toward the center of the battery support with enough pressure to disengage the flanges that hold the battery support in place.
3. Gently pull the support downward until it is completely free from the terminal housing.
4. Remove the old battery from the support.
5. Install the new battery in the support making sure that the positive end of the battery matches the positive end of the support (+ to + and − to −).
6. Reinsert the battery support into the terminal. A slotted guide along one side of the battery support ensures that the support is inserted correctly. The slotted guide must be facing away from the terminal case when you reinsert the support (otherwise the support will not fit back into the terminal).

THERMAL PRINTER PAPER

The optional thermal printer mechanism uses a thermal printing paper that is manufactured specifically for use by the HP 262x family of terminals. You can purchase it through your local HP Sales and Service Office using the following nomenclature and HP part number:

1 Box (24 rolls) Thermal Paper, HP Part No. 9270-0638

It is recommended that you only use HP Thermal Paper in your terminal. If you have an HP Warranty and Service Contract, you MUST use only HP Thermal Paper in order to maintain a valid contract. HP Warranty and Service Contracts are available through your local HP Sales and Service Office.

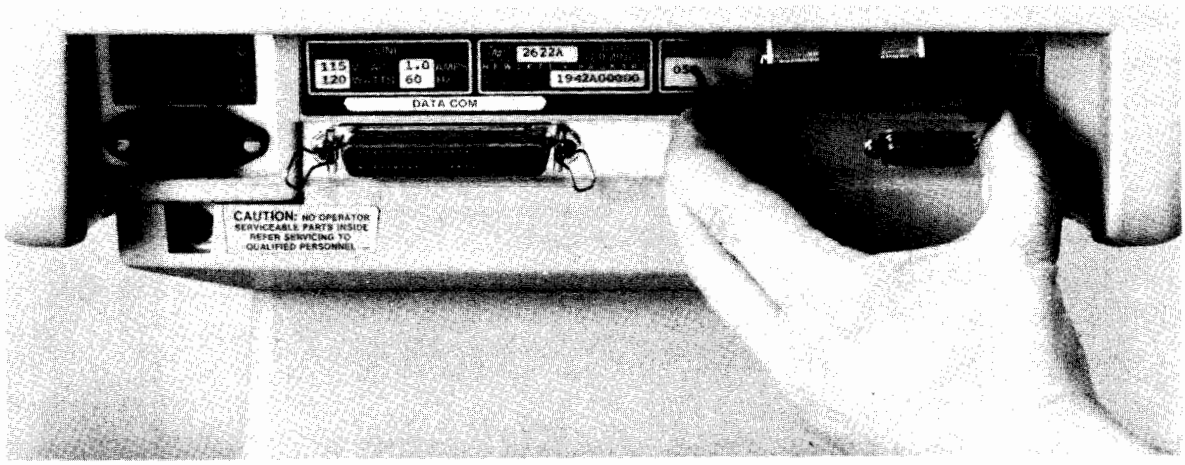


Figure 9-2. Removing the Battery

Paper Loading

The printer mechanism is shown in figure 9-3.

Load a roll of thermal paper into the printer as follows:

1. Lift the top cover of the printer mechanism. An illustration of the correct paper position and flow is embossed on the underside of this cover.
2. Press the latch (figure 9-3) toward the front of the terminal to release the latching frame. Lift the hinged latching frame to its forward position.
3. Remove any paper remaining in the printer.
4. The center paper core is held in place by a metal rod inserted through the center of the core. Grasp the core and lift forward and upward along the guide slots to remove the core and rod.
5. Remove the rod from the old core and insert the rod through the core of a new roll of paper.

   The HP Thermal Paper is coated with print material on one side and must be inserted into the printer correctly to produce the print image. The paper must feed toward the front of the terminal from the underside of the paper roll (see the embossed illustration on the top cover).
6. Place the ends of the metal rod into the guide slots on either side of the print mechanism and press downward and then toward the back of the terminal until the rod snaps into place.
7. Feed the leading edge of the paper through the latching frame (between the latching frame and the clear plastic guide window). Be careful not to sharply strike the print head because damage may result.
8. Lower the latching frame without locking it into place.
9. Align the sides of the paper with the guide lines embossed on each side of the guide window.
10. Each new roll of HP Thermal Paper has a glue spot near the leading edge of the roll that holds the paper roll intact during shipment. You must not allow the print head to come in contact with this glue spot.

    Feed approximately 12 inches of paper through the latching frame so that the glue spot is beyond (outside) the print head and guide window.
11. Press the latch down until it locks into place with an audible click.
12. Tear off the excess paper using the edge of the guide window as a cutting edge.
13. Close the top cover securely and press RETURN.

Note that if subsequent printer operations produce no image on the paper, the paper has probably been installed with the wrong side facing the print head. An image can be printed only on one side of HP Thermal Paper.

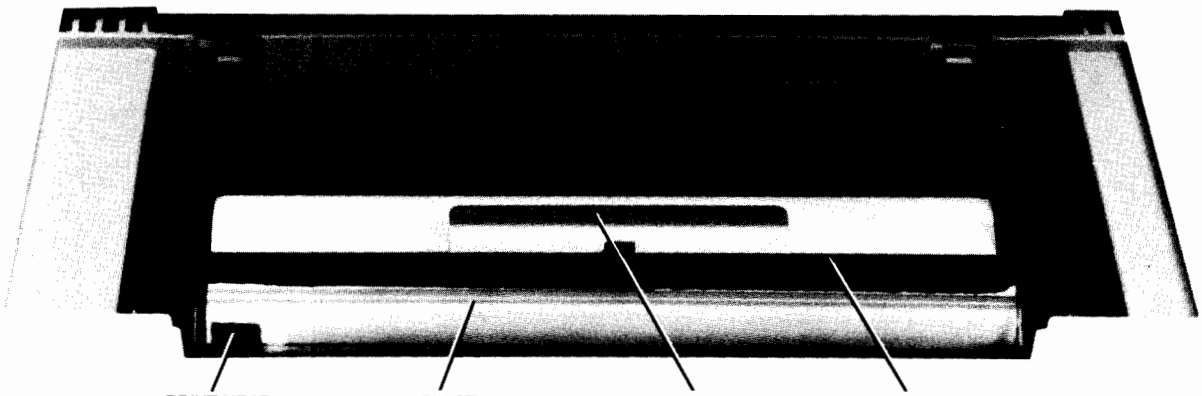


Figure 9-3. Printer Mechanism

Graphics Control Functions

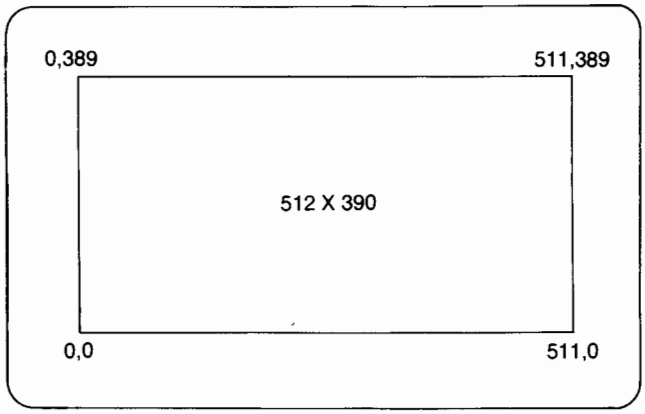
SECTION X

INTRODUCTION

This section contains a description of the terminal's graphics functions and how they are used. The information and the examples are intended for use in developing programs to control the graphics functions. Additional information on how to use the graphics features from the keyboard is contained in the User's Manual.

GRAPHICS DISPLAY

You can display graphics data by addressing points in a 512 by 390 array.



The graphics and alphanumeric data are displayed in the same area on the screen but are stored in separate RAM memories. This allows you to read or modify graphics and alphanumeric data separately.

NOTE

Display enhancements (blinking, inverse video, half-bright, and underline) will affect the display of graphics data. The display enhancements affect alphanumeric character positions on the screen. Since the graphics data is displayed using the same screen dots as the alphanumeric data, it is also affected by the enhancements. The data in the graphics display memory is unchanged.

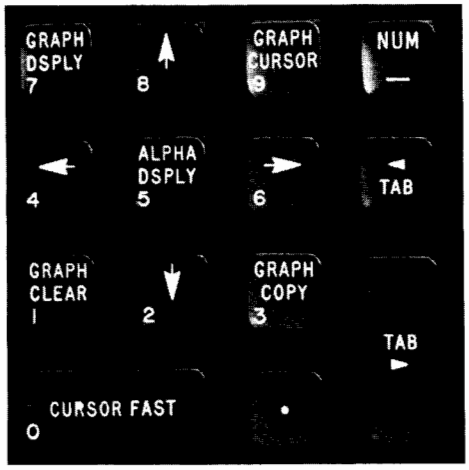


Figure 10-1. Location of Graphics Keys

KEYBOARD GRAPHICS FUNCTIONS

All of the graphics functions commands can be entered from the terminal keyboard by the operator. Some of the functions are available through a special set of graphics control keys located to the right of the normal ASCII character set (see figure 10-1). Table 10-3 contains a list of the keys and a description of their functions. The functions performed by these keys can be used in both local and remote operation. This allows a combination of operator and program control of graphics functions to be used to make maximum use of the terminal's capabilities. Detailed information for using the graphics control keys is contained in the User's Manual.

Each key (except the two TAB keys and the decimal point key) performs two functions. The NUM key controls whether or not the "10-key" pad performs the graphics functions labeled on the keys or the numeric functions labeled in the lower-left corner of each key. When the terminal is turned on initially, or after a hard reset SHIFT CTRL RESET, the keypad is in graphics mode. This means that the graphics functions labeled on the keys are in effect. To change to the numeric functions, press SHIFT NUM. To change back to graphics functions, press SHIFT NUM again. As you can see, SHIFT NUM toggles the keypad functions between graphics and numerics. The only graphics keys that repeat when held depressed are those that control the graphics cursor.

When the keypad is in graphics mode, the minus function on the NUM, and the TAB◄, TAB►, and . keys are disabled.

Table 10-1. Graphics Control Keys

KEY	DESCRIPTION
[GRAPH CURSOR]	Toggles the graphics cursor on and off.
◄ ► ▲ ▼	Move the graphics cursor. More than one can be pressed for diagonal motion.
[CURSOR FAST]	Speeds up the graphics cursor if pressed in conjunction with the cursor keys. The rate returns to normal when released.
[GRAPH DSPLY]	Toggles the graphics display, to inhibit the graphics image without erasing.
[ALPHA DSPLY]	Toggles the alphanumeric displays.
[GRAPH CLEAR]	Erases the graphics image memory.
[GRAPH COPY]	Copies graphics memory to the specified "to" devices.
[NUM —]	Toggles the function of the keypad between graphics and numerics when SHIFT is pressed simultaneously. Unshifted in numeric mode, the key is used to display a dash (—) character.

BIT 4321 \ 765	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	¦
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Parameters Commands

BIT 7 6 5 4 3 2 1

0 0	Control Code
0 1	Parameter
1 0	Command and Terminate Sequence
1 1	Command and Continue Sequence

PROGRAMMABLE GRAPHICS FUNCTIONS

Graphics functions are controlled by parameterized escape sequences. All graphics escape sequences begin with Esc *. The third character, always lower case, selects the type of graphics sequence. Table 10-2 lists the types of graphics sequences. For example, Esc * p specifies a plotting sequence.

Subsequent characters in the control sequence are read as either parameters or commands, depending on the location of the character in the ASCII table.

Table 10-2. Summary of Graphics Sequence Types

ESCAPE SEQUENCE	DESCRIPTION
Esc * d	Display Control
Esc * l	Labeling
Esc * m	Drawing Mode
Esc * p	Vector Plotting
Esc * s	Graphics Status
Esc * t	Compatibility Mode

Control Codes

Control codes are generally ignored, with the exception of the ESCAPE character (Esc). If an Esc character is detected and the previous graphics control sequence has not been properly terminated with a "Z" *or some other valid upper case character*, the Esc will cause the execution of the previous sequence to be terminated. The new escape sequence will then be executed.

Commands

Graphics commands come from columns 4–7 of the ASCII table, the upper and lower case letters (A–Z and ^). Both upper and lower case commands execute the same function. *Upper case letters terminate the sequence and cause it to be executed. You can use more than one command in a sequence.*

Graphics sequences can be any length. (The terminal ignores CR and LF characters in the middle of graphics sequences.) For example, to plot a figure containing 100 points the escape sequence could appear as follows:

```
Et * p a <x1,y1> . . . <x100,y100>Z
```

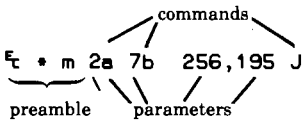
This could cause problems if an error occurs and the system tries to report it in the middle of a long sequence. Since most systems use upper case characters for messages, the first character of the message would end any graphics sequence that might be in progress. Letters that have not been assigned a function for a particular graphics sequence are treated as NOPs and if they are lower case, are ignored. If upper case, they will end the sequence. The letter z has been defined as a NOP in all sequences so that a capital Z can always be used to end a graphics escape sequence.

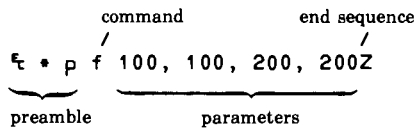
Parameters

Parameters come from columns 2 and 3 of the ASCII table (SPACE through ?). Most parameters are simply the ASCII numeric characters used to represent data coordinates or to select one of several settings. Binary formatted data is generated by appending the bits 0 1 to five bits of binary data. Note that in binary formats, spaces are treated as data and are not ignored or used as delimiters. Both ASCII and binary data formats are described later in this section.

Parameters precede their associated commands (postfix notation). The most frequently used parameters are vector data. Refer to the discussion of Vectors for additional information on parameters used to define vector operations.

Examples:





The programmable graphics functions are organized into five major groups.

- Graphics Display Control
- Plotting
- Graphics Text
- Compatibility Mode

The remainder of this section contains descriptions of each of these functional groups.

GRAPHICS DISPLAY CONTROL

Graphics display control is made up of the functions used to control the graphics cursor, the portion of the graphics memory that is currently being displayed, or the state of the graphics memory. These functions are as follows:

- Graphics Cursor Control
- Graphics Memory Control

Table 10-3 lists the escape sequences for each of the graphics display control functions.

Table 10-3. Graphics Display Control Functions

FUNCTION	CODE	DESCRIPTION
Graphics Cursor Control		
Cursor On	Et*dk	Turn on the graphics cursor.
Cursor Off	Et*dl	Turn off the graphics cursor.
Move Absolute	Et*d<x,y>o	Position the graphics cursor.
Move Relative	Et*d<x,y>p	Position the graphics cursor.
Graphics Memory Control		
Clear Memory	Et*da	Turn off all dots in graphics memory.
Set Memory	Et*db	Turn on all dots in graphics memory.
Display On	Et*dc	Enable the graphics display.
Display Off	Et*dd	Inhibit the graphics display.

Graphics Cursor Control

A separate graphics cursor is available for use in locating points in the graphics display. The graphics cursor is used by the terminal operator to input position data or to interact with a graphics application program.

GRAPHICS CURSOR ON/OFF. The graphics cursor is initially off (power on or full reset). Turning the cursor on or off does not effect the data in graphics memory.

The graphics cursor may be toggled on and off by pressing the [GRAPH CURSOR] key on the graphics/numeric pad.

Programmatically, you can toggle the cursor:

Graphics Cursor On: `Et*dk`

Graphics Cursor Off: `Et*dl`

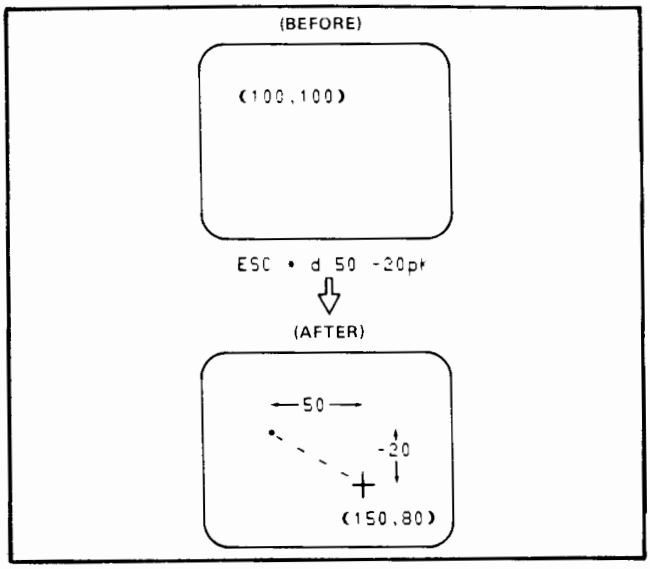
GRAPHICS CURSOR POSITIONING. The graphics cursor is initially at position (0,0) after power on or a full reset. The cursor can be positioned (even if it is not turned on) using either absolute or relative coordinates. In the following sequences X and Y give the new cursor position. Refer to Vectors for a discussion of absolute and relative coordinates.

Position Graphics Cursor Absolute: ESC * d <X,Y> o

Position Graphics Cursor Relative: ESC * d <X,Y> p

You may position the graphics cursor from the graphics/numeric keypad by pressing ◄, ►, ▲, or ▼. Pressing two keys simultaneously will cause diagonal movement. The [CURSOR FAST] key may be pressed simultaneously to speed up cursor positioning.

Example: The cursor is currently at position 100,100 and off. Move it 50 units to the right and 20 units down from its current position and turn it on.



Graphics Memory Control

The graphics display can be turned on or off or the entire memory can be set to all ones (dots on) or all zeros (dots off).

GRAPHICS DISPLAY ON/OFF. The graphics display and graphics cursor can be turned on or off. The data in the graphics memory is unaffected.

From the graphics/numeric keypad, pressing [GRAPH DSPLY] toggles the graphics display on and off:

Programmatically:

Graphics Display On: ESC * dc

Graphics Display Off: ESC * dd

GRAPHICS DISPLAY SET/CLEAR. The graphics data currently displayed on the screen can be set to all ones (a white screen) or cleared to all zeros (a black screen).

Clear Graphics Memory: ESC * da

Set Graphics Memory: ESC * db

You may clear graphics data on the screen from the keyboard by pressing [GRAPH CLEAR] on the graphics/numeric pad.

INSERTING DELAYS IN GRAPHICS OPERATIONS

Certain graphics operations are executed only at the end of a frame (drawing of the display). These operations take approximately 16 ms and can be used as delays to slow down or synchronize changes in the display.

OPERATION	CODE
Graphics Display On	ESC * dc
Graphics Display Off	ESC * dd
Alphanumeric Display On	ESC * de
Alphanumeric Display Off	ESC * df

GRAPHICS DRAWING MODE PARAMETERS

There are several drawing parameters that can be set to allow a wide variety of drawing capabilities. These parameters select whether data will be stored in the graphics memory as 1's or 0's, define line or area patterns to be used when drawing vectors, position the relocatable origin, and define graphics text settings.

Graphics drawing control sequences begin with ESC * m followed by one or more of the drawing parameters. Table 10-4 lists the mode control commands.

Drawing Modes

Vectors can be drawn by setting, clearing, or complementing the data in the graphics memory. Normally the memory is cleared and vectors are drawn by setting selected bits to make white lines on a dark screen. If instead you want black vectors on a white screen, you can begin by setting memory (refer to the Set Memory command), select a clear or complement line type and draw dark vectors (refer to the example that follows). Figure 10-2 illustrates the various drawing modes.

Table 10-4. Graphics Mode Commands

ESC * m <parameters>	
PARAMETERS	**DESCRIPTION**
a	select drawing mode
b	select line type
c	define line pattern
d	define area pattern
e	area fill, absolute
f	area fill, relocatable
j	set relocatable origin
k	set relocatable origin to current pen position
l	set relocatable origin to cursor position
m	set graphics text size
n	set graphics text direction
o	turn on charcter slant
p	turn off character slant
q	set text origin
r	set graphics defaults
z	NOP

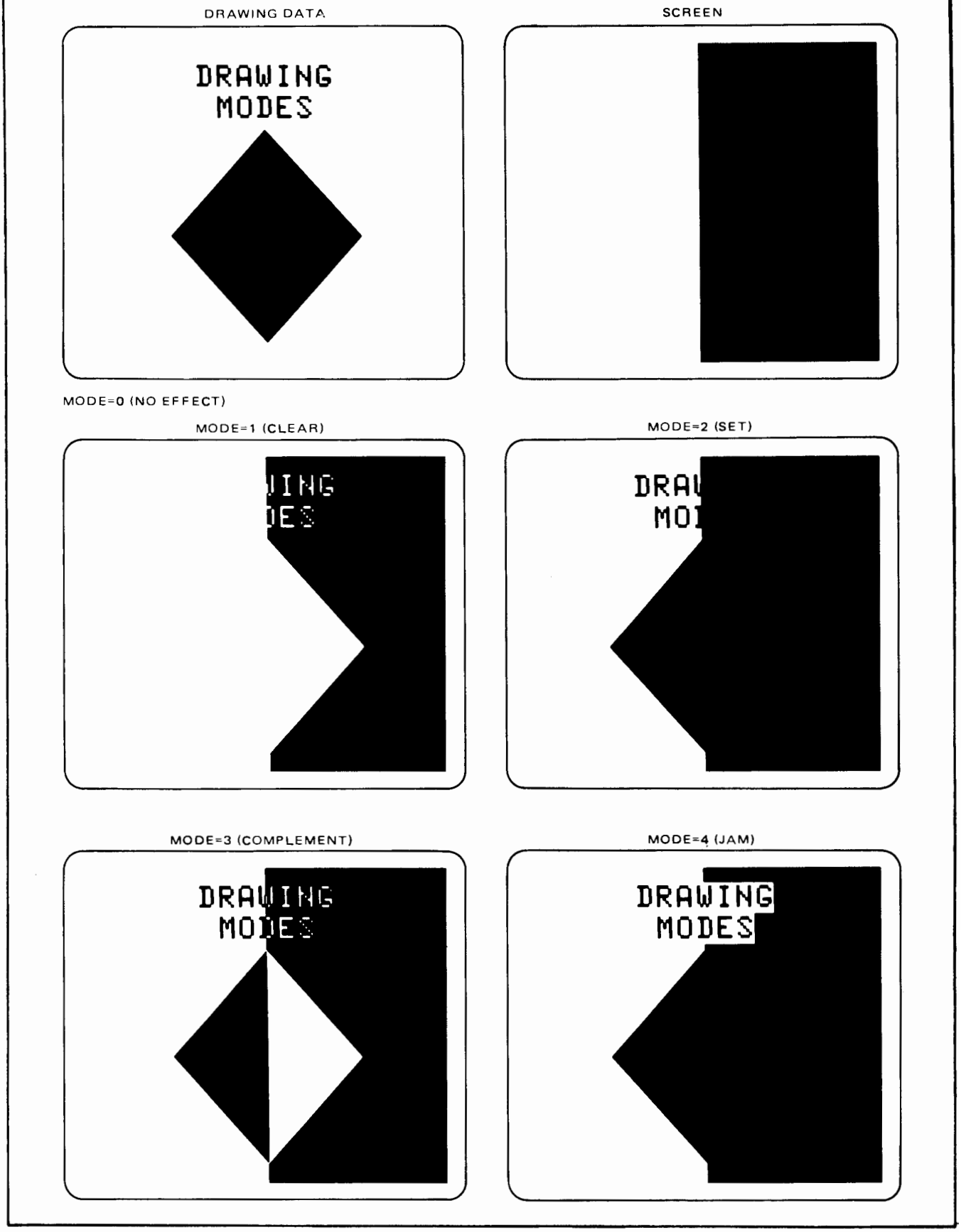


Figure 10-2. Examples of Drawing Modes

Set Drawing Mode: `Esc * m <parameter> a`

where: `<parameter>` is

1 Clear (turn off graphics bits).
2 Set (turn on graphics bits).
3 Complement (toggle the graphics bits).
4 Jam (turn bits on or off according to the data).

CLEAR MODE. Clear mode causes selected display bits to be turned off. The "selected bits" are those that are "on" in the line pattern. If a solid line type (the default) has been selected, all of the bits in a vector will be selected. In clear mode this means that all of the dots making up a vector will be turned off. This allows you to draw dark vectors on a white background. Only those bits that are on in the pattern are cleared. Bits that are off in the pattern do not affect the display.

SET MODE. Set mode is similar to clear mode except that the selected bits are turned on instead of off. Only the bits that are on in the line type are affected.

COMPLEMENT MODE. Complement mode causes the selected display bits to change state (on to off, off to on). Again only those bits that are on in the line type or pattern are affected.

JAM MODE. Jam mode differs from the other modes in that both the bits that are on in the line type or pattern and the bits in the pattern that are off affect the display. Jam mode has the effect of overlaying the display with the pattern.

Selective Erase. A vector drawn in set mode can be selectively erased by redrawing it in clear mode. This will cause gaps to occur if the erased line is intersected by other lines. This problem can be overcome by initially drawing the line in complement mode and then redrawing it in complement mode to erase the line. This technique will preserve the original display. Complement mode is useful for drawing and erasing temporary figures.

Example: Select complement mode, draw a vector, and then erase the vector by redrawing.

```
Esc * m 3A          (select complement mode)
Esc * p a f 100,300 300,300Z    (draw vector)
Esc * p a f 100,300 300,300Z    (erase vector)
```

Drawing Patterns

You can select the dot pattern used when drawing vectors or filling rectangular areas. Dotted and dashed lines can be drawn by selecting one of nine predefined line patterns or a user defined line or area pattern. This allows you to use different line patterns to distinguish between groups of plotted data or easily generate shading and cross hatching for use in engineering drawings, graphs or fabric patterns.

LINE TYPE. One of eleven line types can be selected. Once a line type has been selected, all drawing vectors are drawn using that line type. The patterns for the predefined line types are shown in figure 10-3. Refer to the Define Line Pattern command for additional information.

Select Line Type: `Esc * m <line type> b`

where: `<line type>` is

1 Solid line (default)
2 User defined line pattern
3 User defined area pattern
4 Predefined pattern #1
5 Predefined pattern #2
6 Predefined pattern #3
7 Predefined pattern #4
8 Predefined pattern #5
9 Predefined pattern #6
10 Predefined pattern #7
11 Point plot

Point plot causes a single point to be plotted at the coordinates specified by the data. This line type is useful for generating "scattergram" type graphs. If user defined area shading is selected (type = 3) the line patterns used are selected from the eight lines making up the area fill pattern (refer to Define Area Pattern). The display is divided into groups of eight rows and eight columns. Horizontal and vertical lines are drawn using the appropriate row or column from the area pattern. Diagonal lines are drawn using a solid vector.

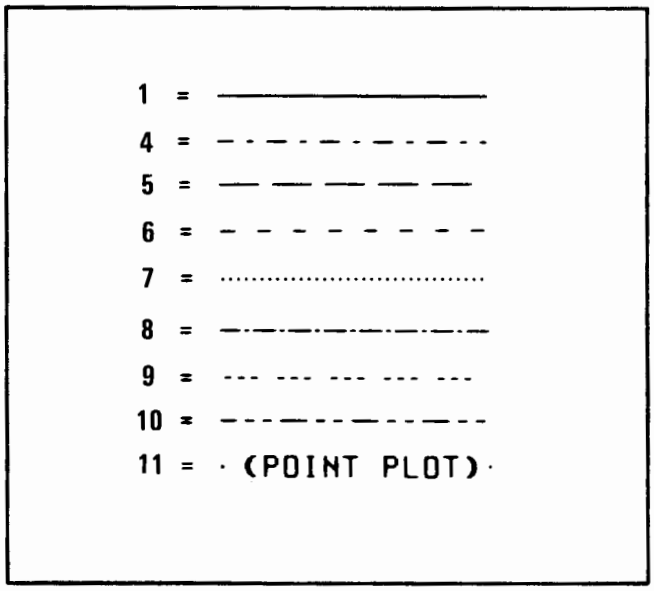
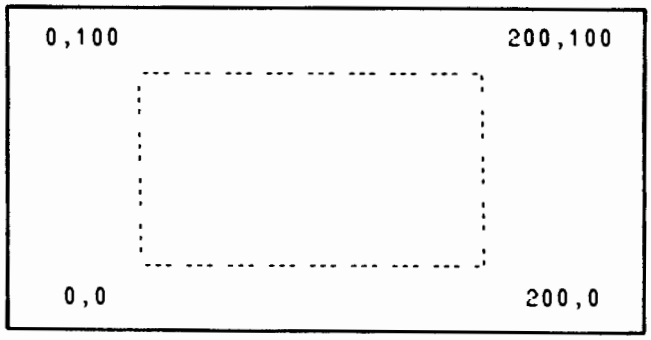


Figure 10-3. Predefined Line Type Patterns

Example: Select line type 9 and draw a figure using the new line type.

```
ESC * m 9 B
ESC * p a 0,0 200,0 200,100 0,100 0,0Z
```



Example: Select the area pattern as the line type.

```
ESC *m51,204,
51,204,51,204,
51,204 D
```
(Defines area pattern shown in figure 10-4b)

```
ESC * m 3 B
```
(Selects the area pattern)

```
ESC *pa2,3,7,3Z
ESC *pa9,3,9,12Z
ESC *pa2,10,7,5Z
```
(Drawn the vectors shown in figure 10-4a)

Drawing vectors (2,3)–(7,3), (9,3)–(9,12), and (2,10)–(7,5) using the area pattern shown in figure 10-4b would result in the drawing shown in figure 10-4a.

Note: Only horizontal and vertical vectors can be defined with an area pattern. All diagonal vectors are drawn as a solid line.

Adjacent horizontal or vertical lines using the user defined line type (type = 2) can be used to create patterns more complicated than those available in an 8×8 area pattern. User defined line and area patterns are described in the following paragraphs.

DEFINE LINE PATTERN. The dot pattern used to draw vectors can be defined programmatically. Once a pattern is defined, you must select the user defined line type (type = 2) using the Select Line Type command (ESC * m 2 B). Figure 10-5 gives examples of line patterns.

Define Line Pattern: `ESC * m <pattern> <scale> c`

where: `<pattern>` is the decimal value 0 to 255) defining an 8-bit binary pattern. For example,
. . . . = 10101010 = 170.

`<scale>` is a scale factor (1 to 16) to be applied to the pattern. For example, with a scale factor of 3 the pattern defined above would be as follows: ... ...

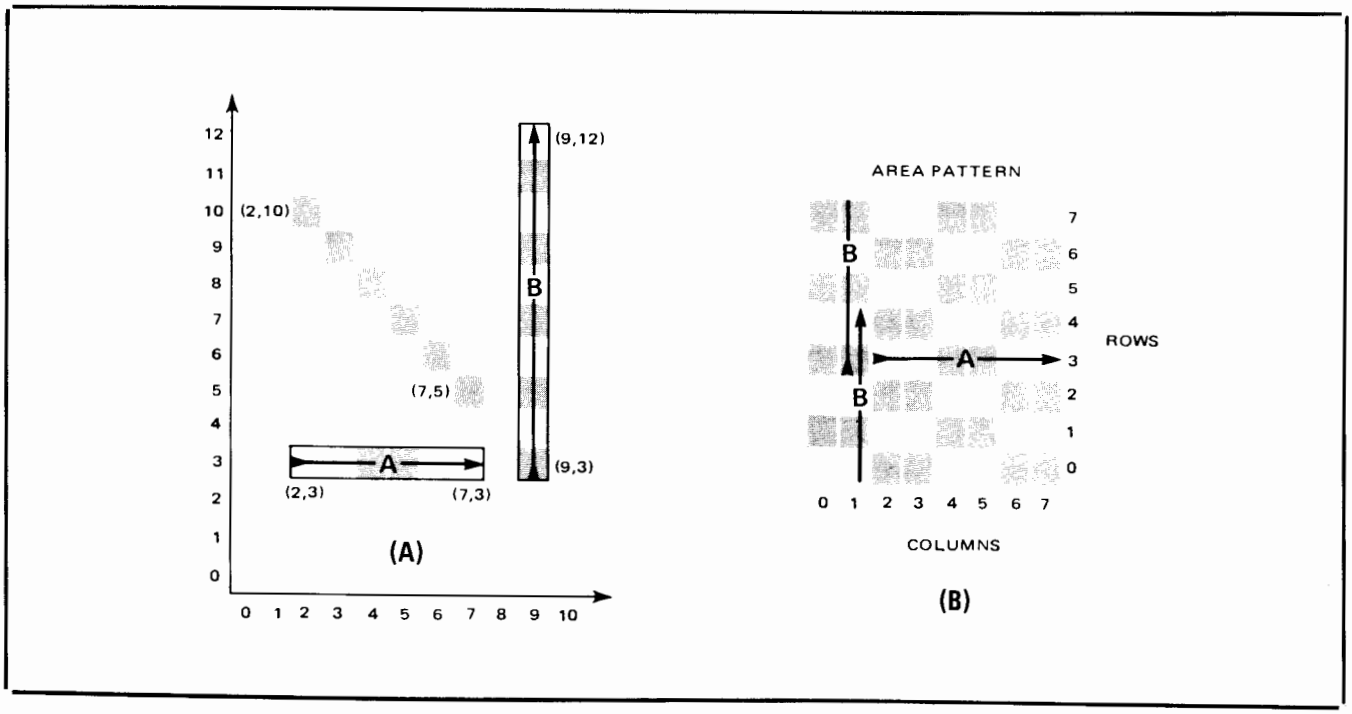


Figure 10-4. Using Area Patterns as Line Types

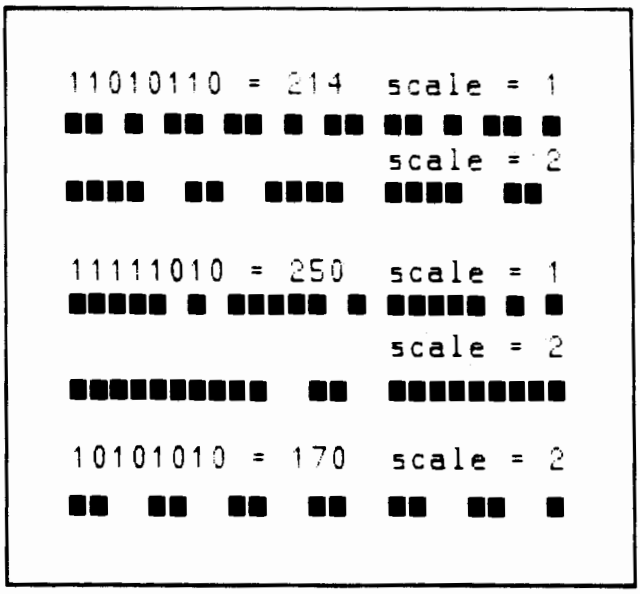


Figure 10-5. Examples of User Defined Line Patterns

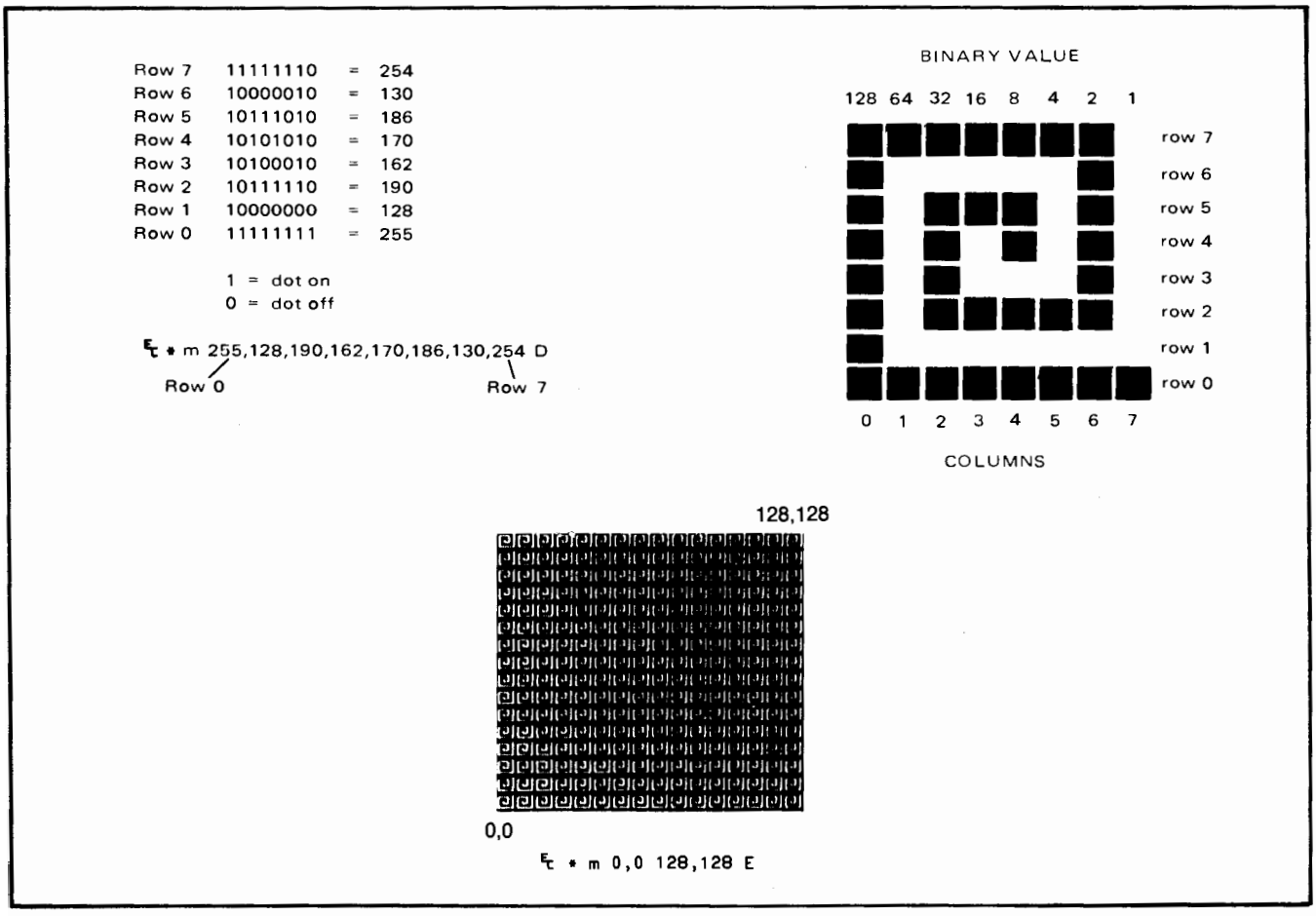
Line patterns too complex to be obtained from an 8×8 area pattern can be generated by plotting a series of lines and varying the patterns used for successive lines. Complex patterns such as those used in weaving can be generated easily using this technique.

Example: Define a pattern to generate the following vector:

```
**********oo**oo**********oo**oo
pattern = 11111010 = 250
scale   = 2
Et * m 250 2 C
```

DEFINE AREA PATTERN. An 8×8 pattern can be defined for use in filling rectangular areas. The pattern can also be used to provide line patterns for horizontal or vertical lines when the area pattern is selected as a line type (type = 3). (Refer to Define Line Type.) Irregular shapes can also be built up by selecting the area shading pattern and then using successive lines.

The area pattern is defined using 8 parameters, one for each of the rows in the pattern. Each parameter is a decimal number (0 to 255) representing an 8-bit binary pattern. Refer to Define Line Pattern for additional information. The display is divided up into 8×8 cells. Every point on the display is mapped to a corresponding bit in the pattern. Drawing horizontal or vertical lines causes the corresponding row or column of the pattern to be used as the line pattern. Diagonal vectors will always be drawn using a solid line. Figure 10-6 contains sample area fill patterns.



Define Area Pattern:

```
Ec * m <row 0> <row 1>...<row 7> D
```

where: <row 0> is the 8-bit pattern for row 0

.
.
.

<row 7> is the 8-bit pattern for row 7

A simple checkerboard pattern would be defined as follows:

```
Ec*m 170 85 170 85 170 85 170 85 D
      |                        |
    Row 0                    Row 7
```

Select the area pattern as the line type:

```
Ec * m 3 B
```

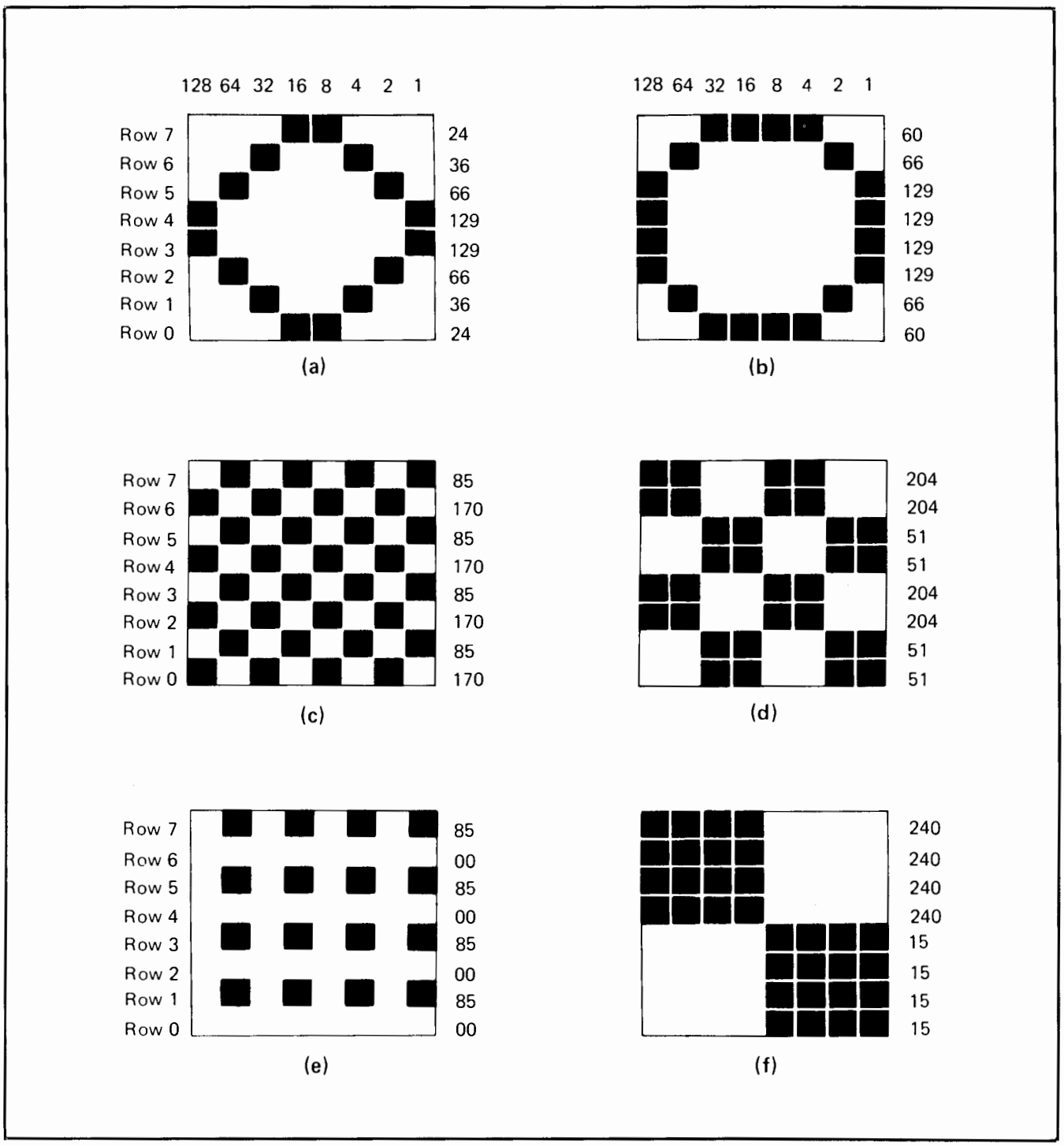


Figure 10-6. Area Pattern Examples

Area Fill

A rectangular area can be filled with a pattern by simply sending the lower left (LL) and upper right (UR) coordinates of the rectangle. The coordinates can be in either absolute or relocatable format. The pattern used is selected by the Line Type command. This allows a choice of predefined and user defined line patterns as well as an 8×8 bit area pattern (refer to Define Area Pattern).

An easy way to selectively erase a portion of the graphics display would be to set the drawing mode to clear, select the solid vector line type (type = 1), and then use the area fill command to select the area to be cleared. Area fill is also useful for shading bar graphs or engineering drawings. The soft keys can be loaded with the proper escape sequences and then triggered to generate area patterns locally using either the current cursor or pen position as a coordinate.

When an area fill is used, the currently selected line type is used for the fill. (Refer to "Drawing Patterns" discussed previously.)

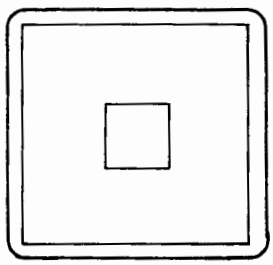
AREA FILL, ABSOLUTE. The absolute area fill command uses the absolute coordinates of the area.

Area Fill, Absolute: `ESC * m <XLL,YLL> <XUR,YUR> e`

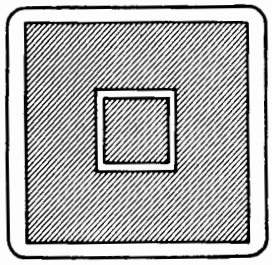
where: `<XLL,YLL>` and `<XUR,YUR>` are the absolute coordinates of the lower left and upper right corners of the area to be filled.

Example: Draw a box and then complement the entire graphics display. Note that repeating the `ESC * m` sequence would restore the original display.

```
ESC*p a 150 150 200 150 200 200 150 200 150 150 F
```



```
ESC * m 3 a 1 b 0 0 511 389E
```



Example: Clear the display area with lower left (LL) coordinates 0,0 and upper right (UR) coordinates 100,100.

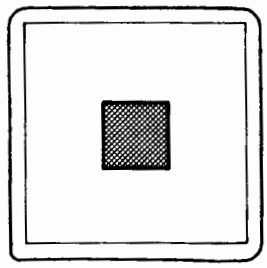
```
ESC * m 1 a 1 b 0,0 100,100 E
```

clear mode — solid line — area to be cleared

If a predefined or user defined line pattern is selected, the area fill command can be used to provide area shading or complex patterns. If the user defined area pattern is selected, the area will be filled with the 8×8 area pattern (refer to Define Area Pattern). (To select line type, refer to "Line Type" on page 10-6.)

Example: Using the area fill pattern shown in figure 10-6a, shade the area with XLL,YLL = 50,50 and XUR,YUR = 100,100.

```
ESC * m 2 a 3 b 24 36 66 129 129 66 36 24 d 50 50 100 100 E
```



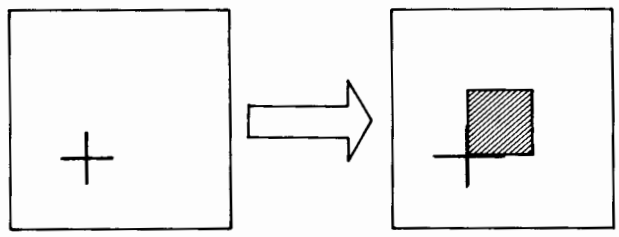
AREA FILL, RELOCATABLE. The relocatable area fill command uses area coordinates in relocatable ASCII format.

Area Fill, Relocatable: `ESC * m <XLL,YLL> <XUR,YUR> f`

where: `<XLL,YLL>` and `<XUR,YUR>` are the relocatable coordinates for the lower left and upper right corners of the area to be filled.

Example: Using the area fill pattern shown in figure 10-6c, shade a 50×50 unit area with the lower left corner at the current cursor position.

```
ESC * m 2 a 3 b 85 170 85 170 85 170 85 170 d 1 0 0 50 50 F
```



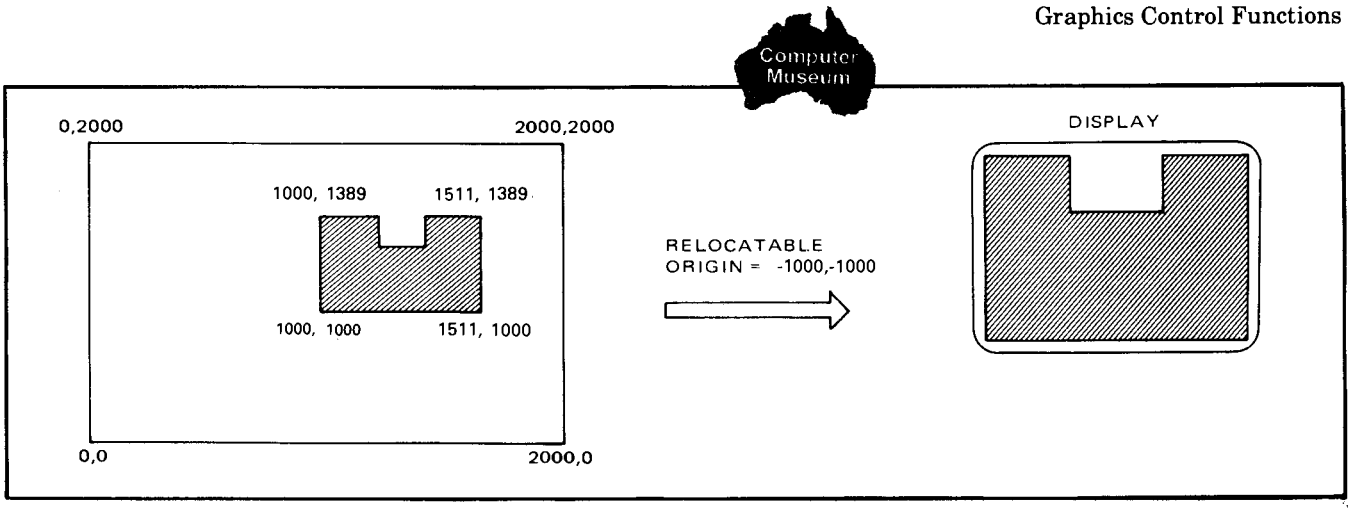


Figure 10-7. Relocatable Origin

Relocatable Origin

The relocatable origin allows you to use one set of data and drawing commands to display a figure at several different positions on the screen. (See the resistor example under ASCII Relocatable Format.)

You can also display portions of a figure that is too large to fit on the screen. You can create a "window" that can be positioned to display any 512 by 390 unit portion of the figure. The value of the relocatable origin is added to the relocatable data to obtain the coordinates used to draw the data. Figure 10-7 illustrates the effect of a Relocatable Origin on the display.

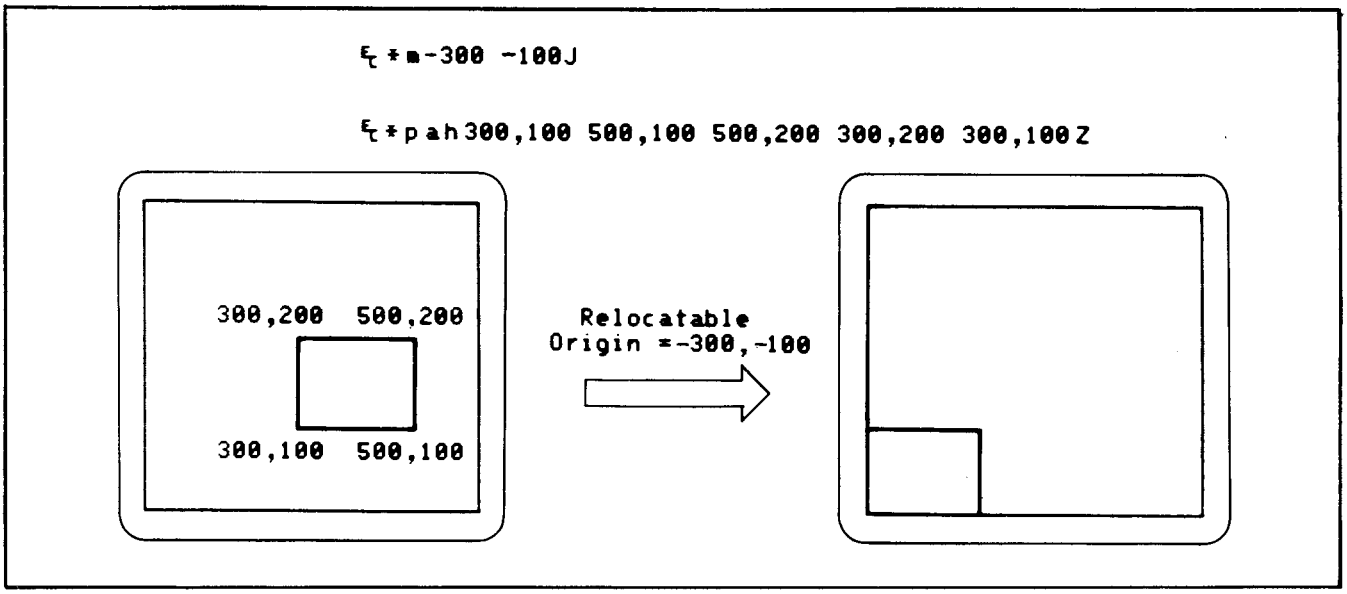
This technique eliminates the need to check boundary conditions or compute new data in order to display the desired portion of the figure and then send the unchanged figure data to the terminal. The terminal will then automatically select and adjust the "window" data.

SET RELOCATABLE ORIGIN, ABSOLUTE. The relocatable origin can be set to any absolute coordinates using ASCII absolute format (−16348 to 16383).

Set Relocatable Origin Absolute: `ESC * m <X,Y> J`

where: `<X,Y>` are the x and y coordinates in ASCII absolute format.

Example: Set the relocatable origin to display the box in the figure so that the box is positioned at the lower left corner of the display.



SET RELOCATABLE ORIGIN TO CURRENT PEN POSITION. The relocatable origin can be set to the current pen position.

Set Relocatable Origin
To Current Pen Position: ESC * m k

SET RELOCATABLE ORIGIN TO GRAPHICS CURSOR POSITION. The relocatable origin can be set to the current graphics cursor position.

Set Relocatable Origin
To Graphics Cursor Position: ESC * m l

Selecting The Graphics Default Parameters

Graphics parameters can be set to their default (power on or full reset) values. Table 10-5 lists the various parameters and their default values. Additional information can be found under the discussions of the individual parameters.

Set Graphics
Default Parameters: ESC * m r

The current graphics mode and settings can be obtained with graphics status requests. Graphics status requests are described in Section VII, Status. It may be desirable to reselect graphics settings before you send graphics data to the terminal.

Table 10-5. Graphics Parameter Default Values

PARAMETER	DEFAULT VALUE
Pen Condition	up
Line Type	1 (solid)
Drawing Mode	set
Relocatable Origin	0,0
Text Size	1
Text Direction	1
Text Origin	1 (left, bottom justified)
Text Slant	0 (off)
Graphics Text	off
Graphics Video	on
Alphanumeric Video	on
Graphics Cursor	off
Alphanumeric Cursor	on
Compatibility Mode	
Page Full Straps	0 (out)[1]
GIN Strap	0 (CR only)[1]

[1]See table 10-3.

PLOTTING SEQUENCES

All vector plotting sequences are initiated by ESC * p. Table 10-6 lists the commands that can be used within a plotting sequence.

Table 10-6. Graphics Plotting Control Functions

ESC * p <parameters and data>	
PARAMETER	**DESCRIPTION**
a	lift the pen
b	lower the pen
c	use graphics cursor position as new point
d	draw a single dot at the current pen position
e	set relocatable origin = current pen position
f	use ASCII absolute format
g	use ASCII incremental format
h	use ASCII relocatable format
i	use binary absolute format
j	use binary short incremental format
k	use binary incremental format
l	use binary relocatable format
z	NOP/synch

After ESC * p has been sent, the drawing format is normally specified before data is sent.

If no format is specified, ASCII absolute is assumed. There is no explicit draw vector command. When enough parameter bytes to specify a single end point have been received (the number depends on the format used), the pen is moved from its current position to the new end point. (See figure 10-8.) If the pen is down, a vector will be drawn. If the pen is up, the pen is moved to the new point (without drawing a vector) and lowered. The new end point becomes the current pen position.

Note that if a parameter byte is lost or garbled in transmission, all following end points will be improperly read. To minimize data errors caused by the loss of a data byte, any command can be used to reset the parameter count and restore synchronization. Nops (z), redundant format, or pen down commands can also be inserted to insure synchronization if necessary.

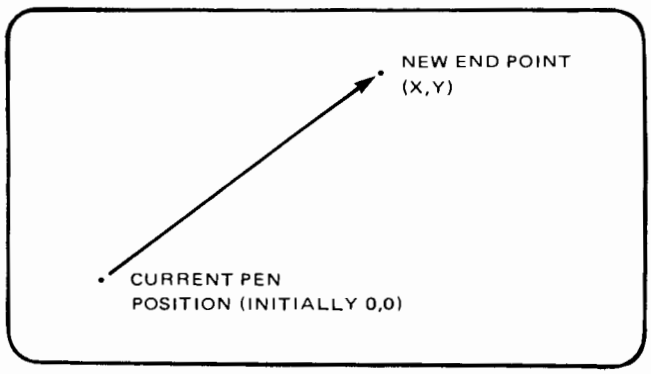


Figure 10-8. Current Pen Position and New End Point

Graphics sequences can extend indefinitely. In general, longer sequences are preferred as they minimize the overhead necessary for a plot sequence. ESC * p <format> must be sent for each series of vectors. As the sequence length decreases, the percentage of preamble characters increases, and the vector drawing rate goes down. The worst possible case would be to send ESC * p <format> for each vector; approximately 50% of the characters sent would be overhead, reducing vector speed by a factor of 2.

The general format for an absolute plotting sequence is:

```
ESC * p i a <byte1> <byte2> <byte3> <byte4>(z)
<byte1> <byte2> <byte3> <byte4> ...
...<byte1> <byte2> <byte3> <byte4>Z (or any
upper case command)
```

Each block of 4 bytes specifies a single point. The "i" indicates that absolute format is to be used. The "a" raises the pen before it is moved to the point specified by the next four bytes and lowered. A NOP (z) can be added to insure synchronization, if necessary. The lowered pen draws a vector as it moves to the next point, and so on. The upper case "Z" terminates the plotting sequence.

The vector end point formats allow the pen to be moved completely off the screen (an absolute coordinate of 1000, for example). The actual range of the pen position can be from −16384 to 16383. Vectors that extend beyond the screen are clipped so that they will not wrap around.

Pen Control

The terminal uses the concept of a "pen" in drawing vector data. The pen can be lifted or lowered as well as be positioned using absolute or relative coordinates. For example, the pen is lifted, moved to a starting coordinate, lowered and moved to an endpoint to draw a line. The pen is initially in the up state and positioned at absolute coordinates 0,0 following power up or a full reset. If the pen is raised and coordinates given, the pen is moved to the coordinates and then lowered. The pen is normally left in the down position.

Raise Pen: ESC * p a

Lower Pen: ESC * p b

Vectors

Graphic data is made up of vectors. Each vector is specified by the current graphic starting point and an end point. The current graphic starting point is one of the following:

0,0 Initial starting point

Last point defined by the graphics cursor (ESC * p c).

Last point defined by data in a draw or move command (ESC * p f/g/h/i/j/k/l).

Graphic points are specified in one of following formats:

- ASCII Absolute
- ASCII Incremental
- ASCII Relocatable
- Binary Absolute
- Binary Incremental
- Binary Short Incremental
- Binary Relocatable

If no format is specified in the graphic command, ASCII absolute format is assumed. More than one point can be given in a command. This minimizes communications overhead. Tables 10-7, 10-8 and 10-9 provide a reference for computing data bytes used in the various vector formats.

ASCII Formats

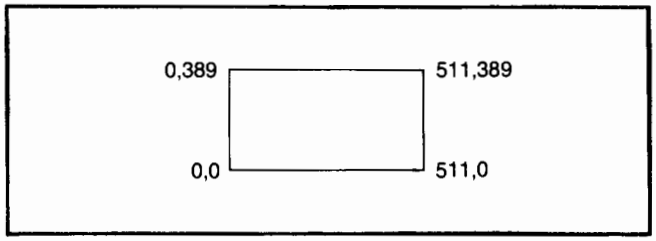
In the ASCII formats, coordinates are specified with ASCII characters 0 through 9. This means that numeric characters generated by a simple print statement can be used to specify X,Y pairs. The first value is used as the X coordinate, and the second as the Y coordinate.

Spaces or commas must be used to delimit the X and Y values. Excess delimiters are ignored. Digits following a decimal point are ignored (i.e. 123.456 is read as 123).

Exponential notation cannot be used. Consequently, the values must be in integer form. The number of bytes necessary to specify a single end point depends on the magnitude of the values.

ASCII ABSOLUTE FORMAT. The values used in the ASCII absolute format can range between −16384 and 16383. Note that only points where X is the range 0 to 511 and Y is the range 0 to 389 will be visible on the screen. The following example draws vectors around the perimeter of the screen:

```
ESC * p a 0,0 511,0 511,389,0,389,0,0Z
```

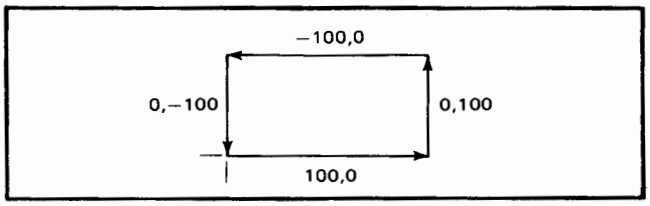


Since no format is indicated, ASCII absolute is assumed. The "a" raises the pen, which is moved to (0,0) and lowered. Vectors are then drawn to (511,0), (511,389), (0,389), and back to (0,0). (Note that the values are delimited by spaces

or commas. The upper case Z [a nop] terminates the sequence. Imbedded carriage return and line feed characters are ignored.)

ASCII INCREMENTAL FORMAT. In the ASCII incremental format you can specify a delta X and a delta Y. These values are added to the current pen position to obtain a new end point. The first value is read as delta X and the second as delta Y. For example to draw a square 100 units on a side, the following sequence could be used:

```
ESC * p g 100 0 0 100 -100 0 0 -100 Z
```



Beginning at the current pen position, a series of vectors is drawn by moving the pen 100 units to the right, up 100 units, left 100 units, and finally down 100 units. The same figure could have been drawn at any screen location by first positioning the pen to the desired starting point before sending the drawing sequence.

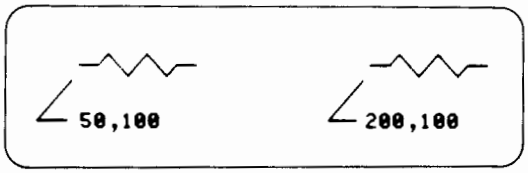
ASCII RELOCATABLE FORMAT. The ASCII relocatable format allows you to use a relocatable origin to be added to the incoming X and Y coordinate values. The resultant values are then treated as absolute coordinates by the terminal. The relocatable format allows you to use absolute data as if it were incremental by merely changing the relocatable origin. For example, symbol elements specified in absolute coordinates can be drawn in different locations as shown in the following example.

Example: Draw a resistor symbol stored in absolute coordinates at screen locations 50,100 and 200,100.

```
Resistor Data = 0,10
                10,10
                15,15      0,20    60,0
                25,5
                35,15
                45,5
                50,10      0,0     60,0
                60,10

ESC*m50,100J
ESC*pah0,10 10,10 15,15 25,5 35,15
                45,5 50,10 60,10Z

ESC*m200,100J
ESC*pah0,10 10,10 15,15 25,5 35,15
                45,5 50,10 60,10Z
```



Binary Format

In binary format all points are sent in a packed binary format. The coordinate values are sent using the bit patterns of the ASCII characters listed in table 10-7. The number of characters required to specify a coordinate depends on the format used. The values for X and Y coordinates can be from −16384 to 16383.

BINARY ABSOLUTE FORMAT. Binary absolute data is plotted with respect to an origin at 0,0. Four bytes are required to specify a single end point. A 10 bit coordinate in the range 0-1023, is sent for both x and y.

The bytes are ordered as follows:

BIT	7	6	5	4	3	2	1	
BYTE 1	0	1	X9	X8	X7	X6	X5	HI X
BYTE 2	0	1	X4	X3	X2	X1	X0	LOW X
BYTE 3	0	1	Y9	Y8	Y7	Y6	Y5	HI Y
BYTE 4	0	1	Y4	Y3	Y2	Y1	Y0	LOW Y

Although it is possible to send coordinates in the range 0 to 1023, only points in the range 0-511 for X, and 0-389 for Y are visible on the screen. Vectors going off the screen are clipped. If the data requires scaling, this must be done before the data is sent to the terminal.

The following example shows how the 4 data bytes are computed. The numbers are converted to the 10 bit binary equivalent. Bits 7 and 6 are set to 01 to indicate a parameter.

```
X = 0 = 00000 00000          Y = 0 00000 00000
        HI X  LOW X                HI Y  LOW Y

        BYTE 1 = 01 00000 = SPACE HI X
        BYTE 2 = 01 00000 = SPACE LOW X
        BYTE 3 = 01 00000 = SPACE HI Y
        BYTE 4 = 01 00000 = SPACE LOW Y


X = 360 = 01011 01000      Y = 180 = 00101 10100
          HI X  LOW X                HI Y  LOW Y

          BYTE 1 = 01 01011 = +  HI X
          BYTE 2 = 01 01000 = (  LOW X
          BYTE 3 = 01 00101 = %  HI Y
          BYTE 4 = 01 10100 = 4  LOW Y
```

An escape sequence to draw a vector from 0,0 to 360,180 is as follows:

```
ESC * p i a SP  SP  SP  SP  +  (  %  4  Z
            \X=0/  \Y=0/ \X=360/ \Y=180/
```

ESC * p selects a plotting sequence. The "i" specifies absolute format. The "a" raises the pen up. The first 4 bytes (all spaces) move the raised pen to 0,0, where it is lowered. The next 4 bytes specify the point 360,180. After the 4th byte is received, the pen is moved to that point, drawing a vector. The upper case "Z" terminates the escape sequence. Note that if spaces are used in the data sequence they are interpreted as data and could result in an improper plot.

BINARY SHORT INCREMENTAL FORMAT. The short incremental format uses two bytes to specify a delta X and a delta Y in the range −16 to +15. The five least significant bits are interpreted as a signed, two's complement number. This number is added to the current pen position to obtain the new end point. The data bytes are ordered as follows:

```
BIT     7   6   5   4   3   2   1

BYTE 1  0   1   <    DELTA X    >
BYTE 2  0   1   <    DELTA Y    >
```

The following example illustrates the computation and use of the short incremental format:

```
DELTA X = -12 = 10100    DELTA Y = 6 = 00110
BYTE1 = 01 10100 = 4   DELTA X
BYTE2 = 01 00110 = &   DELTA Y
```

The following sequence moves the pen to 360,180 in absolute format, then draws a vector to X = 360−12 = 348, y = 180+6 = 186.

```
               absolute
EC * p i a + ( % 4 j 4 & <byte1><byte2>...Z
      Short Incremental   Short Incremental vectors
```

BINARY INCREMENTAL FORMAT. Incremental is similar to short incremental, but with a larger range. Using six bytes, delta X and Y can range from −16384 to +16383.

```
BIT   7 6  5     4     3     2     1

BYTE1 0 1  DX14  DX13  DX12  DX11  DX10  HI  DELTA X
BYTE2 0 1  DX9   DX8   DX7   DX6   DX5   MID DELTA X
BYTE3 0 1  DX4   DX3   DX2   DX1   DX0   LOW DELTA X

BYTE4 0 1  DY14  DY13  DY12  DY11  DY10  HI  DELTA Y
BYTE5 0 1  DY9   DY8   DY7   DY6   DY5   MID DELTA Y
BYTE6 0 1  DY4   DY3   DY2   DY1   DY0   LOW DELTA Y
```

The following example shows how incremental data bytes are generated.

```
DELTA X = -400 = 11111   10011    10000
                 HI DX   MID DX   LO DX

DELTA Y = 100 = 00000   00011    00100
                HI DY   MID DY   LO DY

BYTE 1 = 01 11111 = ?        HI  DELTA X
BYTE 2 = 01 10011 = 3        MID DELTA X
BYTE 3 = 01 10000 = 0        LO  DELTA X

BYTE 4 = 01 00000 = space    HI  DELTA Y
BYTE 5 = 01 00011 = #        MID DELTA Y
BYTE 6 = 01 00100 = $        LO  DELTA Y
```

Table 10-7. Characters Used in Packed Data Formats

ASCII Character	Bit Pattern	ASCII Character	Bit Pattern
SP	01 0 0000	0	01 1 0000
!	01 0 0001	1	01 1 0001
"	01 0 0010	2	01 1 0010
#	01 0 0011	3	01 1 0011
$	01 0 0100	4	01 1 0100
%	01 0 0101	5	01 1 0101
&	01 0 0110	6	01 1 0110
'	01 0 0111	7	01 1 0111
(	01 0 1000	8	01 1 1000
)	01 0 1001	9	01 1 1001
*	01 0 1010	:	01 1 1010
+	01 0 1011	;	01 1 1011
,	01 0 1100	<	01 1 1100
-	01 0 1101	=	01 1 1101
.	01 0 1110	>	01 1 1110
/	01 0 1111	?	01 1 1111

BINARY RELOCATABLE FORMAT. Binary relocatable format specifies absolute X and Y coordinates in the range −16384 to +16383 using 6 bytes. The value specified in the relocatable origin command is taken to be the 0,0 point. The actual screen address is computed by the terminal by adding the relocatable origin to the X,Y pair.

```
BIT     7   6   5    4    3    2    1

BYTE 1  0   1   X14  X13  X12  X11  X10   HI  X
BYTE 2  0   1   X9   X8   X7   X6   X5    MID X
BYTE 3  0   1   X4   X3   X2   X1   X0    LOW X

BYTE 4  0   1   Y14  Y13  Y12  Y11  Y10   HI  Y
BYTE 5  0   1   Y9   Y8   Y7   Y6   Y5    MID Y
BYTE 6  0   1   Y4   Y3   Y2   Y1   Y0    LOW Y
```

The following example shows how relocatable data bytes are computed.

```
RELOC X = -600 = 11111 01101 01000
                 HI X  MID X LOW X

RELOC Y =  200 = 00000 00110 01000
                 HI Y  MID Y LOW Y

BYTE 1 = 01 11111 = ?        HI  X
BYTE 2 = 01 01101 = -        MID X
BYTE 3 = 01 01000 = (        LOW X

BYTE 4 = 01 00000 = space    HI  Y
BYTE 5 = 01 00110 = &        MID Y
BYTE 6 = 01 01000 = (        LOW Y
```

GRAPHICS FUNCTIONS IN DISPLAY FUNCTIONS MODE

The DISPLAY FUNCTIONS key (f7 at the MODES level) can be used to display the graphics escape sequences or the action of graphics control keys. The control sequences are entered into the alphanumeric display each time a command is executed. Table 10-10 lists the graphics control sequences that are generated when DISPLAY FUNCTIONS is on.

Table 10-8. Absolute Format Addressing Bytes

	0	1	2	3	4	5	6	7	8	9
0	■■	■!	■"	■#	■$	■%	■&	■'	■(	■)
10	■*	■+	■,	■-	■.	■/	■0	■1	■2	■3
20	■4	■5	■6	■7	■8	■9	■:	■;	■<	■=
30	■>	■?	!■	!!	!"	!#	!$	!%	!&	!'
40	!(	!)	!*	!+	!,	!-	!.	!/	!0	!1
50	!2	!3	!4	!5	!6	!7	!8	!9	!:	!;
60	!<	!=	!>	!?	"■	"!	""	"#	"$	"%
70	"&	"'	"(	")	"*	"+	",	"-	".	"/
80	"0	"1	"2	"3	"4	"5	"6	"7	"8	"9
90	":	";	"<	"=	">	"?	#■	#!	#"	##
100	#$	#%	#&	#'	#(	#)	#*	#+	#,	#-
110	#.	#/	#0	#1	#2	#3	#4	#5	#6	#7
120	#8	#9	#:	#;	#<	#=	#>	#?	$■	$!
130	$"	$#	$$	$%	$&	$'	$(	$)	$*	$+
140	$,	$-	$.	$/	$0	$1	$2	$3	$4	$5
150	$6	$7	$8	$9	$:	$;	$<	$=	$>	$?
160	%■	%!	%"	%#	%$	%%	%&	%'	%(	%)
170	%*	%+	%,	%-	%.	%/	%0	%1	%2	%3
180	%4	%5	%6	%7	%8	%9	%:	%;	%<	%=
190	%>	%?	&■	&!	&"	&#	&$	&%	&&	&'
200	&(	&)	&*	&+	&,	&-	&.	&/	&0	&1
210	&2	&3	&4	&5	&6	&7	&8	&9	&:	&;
220	&<	&=	&>	&?	'■	'!	'"	'#	'$	'%
230	'&	''	'(	')	'*	'+	',	'-	'.	'/
240	'0	'1	'2	'3	'4	'5	'6	'7	'8	'9
250	':	';	'<	'=	'>	'?	(■	(!	("	(#
260	($	(%	(&	('	((	()	(*	(+	(,	(-
270	(.	(/	(0	(1	(2	(3	(4	(5	(6	(7
280	(8	(9	(:	(;	(<	(=	(>	(?	)■	)!
290	)"	)#	)$	)%	)&	)'	)(	))	)*	)+
300	),	)-	).	)/	)0	)1	)2	)3	)4	)5
310	)6	)7	)8	)9	):	);	)<	)=	)>	)?
320	*■	*!	*"	*#	*$	*%	*&	*'	*(	*)
330	**	*+	*,	*-	*.	*/	*0	*1	*2	*3
340	*4	*5	*6	*7	*8	*9	*:	*;	*<	*=

	0	1	2	3	4	5	6	7	8	9
350	*>	*?	+■	+!	+"	+#	+$	+%	+&	+'
360	+(	+)	+*	++	+,	+-	+.	+/	+0	+1
370	+2	+3	+4	+5	+6	+7	+8	+9	+:	+;
380	+<	+=	+>	+?	,■	,!	,"	,#	,$	,%
390	,&	,'	,(	,)	,*	,+	,,	,-	,.	,/
400	,0	,1	,2	,3	,4	,5	,6	,7	,8	,9
410	,:	,;	,<	,=	,>	,?	-■	-!	-"	-#
420	-$	-%	-&	-'	-(	-)	-*	-+	-,	--
430	-.	-/	-0	-1	-2	-3	-4	-5	-6	-7
440	-8	-9	-:	-;	-<	-=	->	-?	.■	.!
450	."	.#	.$	.%	.&	.'	.(	.)	.*	.+
460	.,	.-	..	./	.0	.1	.2	.3	.4	.5
470	.6	.7	.8	.9	.:	.;	.<	.=	.>	.?
480	/■	/!	/"	/#	/$	/%	/&	/'	/(	/)
490	/*	/+	/,	/-	/.	//	/0	/1	/2	/3
500	/4	/5	/6	/7	/8	/9	/:	/;	/<	/=
510	/>	/?	0■	0!	0"	0#	0$	0%	0&	0'
520	0(	0)	0*	0+	0,	0-	0.	0/	00	01
530	02	03	04	05	06	07	08	09	0:	0;
540	<■	0=	0>	0?	1■	1!	1"	1#	1$	1%
550	1&	1'	1(	1)	1*	1+	1,	1-	1.	1/
560	10	11	12	13	14	15	16	17	18	19
570	1:	1;	1<	1=	1>	1?	2■	2!	2"	2#
580	2$	2%	2&	2'	2(	2)	2*	2+	2,	2-
590	2.	2/	20	21	22	23	24	25	26	27
600	28	29	2:	2;	2<	2=	2>	2?	3■	3!
610	3"	3#	3$	3%	3&	3'	3(	3)	3*	3+
620	3,	3-	3.	3/	30	31	32	33	34	35
630	36	37	38	39	3:	3;	3<	3=	3>	3?
640	4■	4!	4"	4#	4$	4%	4&	4'	4(	4)
650	4*	4+	4,	4-	4.	4/	40	41	42	43
660	44	45	46	47	48	49	4:	4;	4<	4=
670	4>	4?	5■	5!	5"	5#	5$	5%	5&	5'
680	5(	5)	5*	5+	5,	5-	5.	5/	50	51
690	52	53	54	55	56	57	58	59	5:	5;
700	5<	5=	5>	5?	6■	6!	6"	6#	6$	6%
710	6&	6'	6(	6)	6*	6+	6,	6-	6.	6/

Note: ■ indicates a "space" character; every coordinate address must consist of the two characters shown in the table.

Table 10-9. Incremental (Short) Vector Bytes

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Table 10-10. Graphics Control Sequences Used in Record Operations

Key	Sequence	Description
▲ ► ◄ ▼	none	Graphics cursor controls
CURSOR FAST	none	Graphics cursor fast
GRAPH CURSOR	Et * dL	off
	Et * dK	on
GRAPH DSPLY	Et * dC	on
	Et * dD	off
GRAPH CLEAR	Et * dA	
ALPHA DSPLY	Et * dF	off
	Et * dE	on

Figure 10-9 shows the sequences generated when drawing a simple box. The graphics cursor is initially on and positioned at 0,0.

GRAPHICS HARDCOPY OPERATIONS

There are two methods of obtaining a hardcopy of the contents of graphics memory. One method uses the function keys and graphics keys on the keyboard. The other method uses escape sequences which may be coded into a program running on a host computer. The hardcopy may be output from the optional integral printer or from an external printer connected to the external printer port at the rear of the terminal.

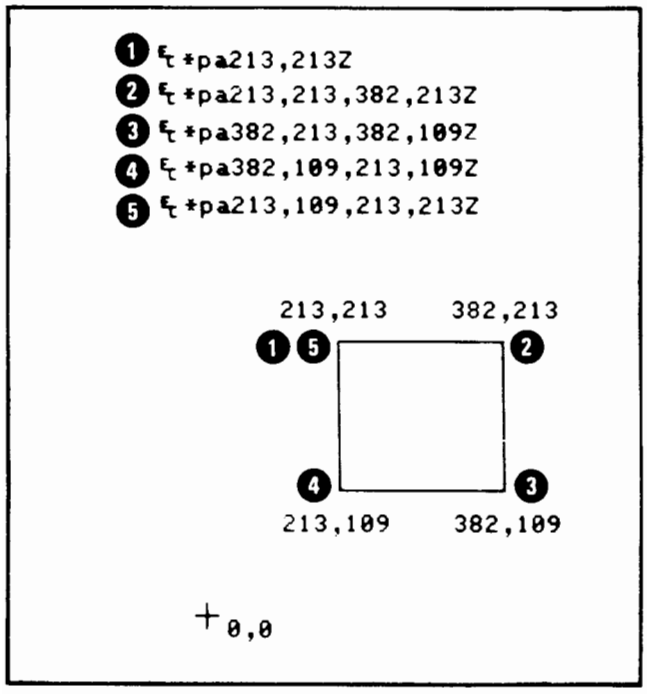


Figure 10-9. Displaying Graphics Sequences

When the optional integral printer is selected as the output device, a dot-by-dot transfer of the graphics memory is made on thermal paper. The image is 14.5 cm by 11 cm (5.75 by 4.25 inches) centered on the paper. The transfer takes approximately 60 seconds.

The external printers that may be used for graphics hardcopy are the HP 2671G and HP 2673A.

Initiating a Transfer from the Keyboard

The [GRAPH COPY] key on the graphics/numeric pad initiates the graphics transfer. The keypad must be in graphics mode for the GRAPH COPY function to be performed. SHIFT NUM toggles the function of the keypad between graphics mode and numeric mode.

The destination(s) may be selected by pressing AIDS, f1 (device control), f3 (to devices). You may then select f2 (TO EXT DEV) and/or f3 (TO INT PRT). TO EXT DEV selects the external printer port, and TO INT PRT selects the optional integral printer.

Using the Et & p Escape Sequences

Coding a program to transfer graphics data either to the integral printer or to the external printer, or both, requires selecting the graphics memory as the source and either or both printers as the destination.

graphics memory as the source:	7s
integral printer as the destination:	6d
external printer as the destination:	4d

Example: Define graphics memory as source and integral printer as destination.

```
Et & p 7s 6D
```

Example: Define graphics memory as source and both printers as destination.

```
Et & p 7s 4d 6D
```

After the source and destination are defined, the transfer if initated by either:

```
Et & p F      (copy file from source to destination)
        or
Et & p M      (copy all from source to destination)
```

Note that an escape sequence is terminated with an uppercase character. Also, you may combine the source and destination assignments and the transfer initiation in one escape sequence:

```
Et & p 7s 4d F
```

GRAPHICS TEXT

Text strings can be written directly into the graphics image memory. An internal character generator converts the ASCII codes into a dot matrix representation which is drawn as vectors. The character set includes upper and lower case (95 characters) and the national characters shown in Appendix B. The national character sets are selected by the Terminal Configuration menu discussed in Section II. The characters will be drawn as a 5 by 7 matrix in a 7 by 10 cell, with descenders for lower case. This character set is in addition to the normal alphanumeric character set. While this character set may seem redundant, it offers the following advantages:

- Characters can be drawn at any dot position, rather than the 24 by 80 alphanumeric character positions.
- Characters can be rotated in multiples of 90 degrees.
- Characters can be scaled in size, from 1 to 8 times.
- Characters can be slanted 45 degrees for an italics-like effect.
- Lines of characters can be right, left, or center justified.

Figure 10-10 shows the graphics character set.

Keyboard Control of Graphics Text

Graphics text can be entered directly from the keyboard. The backspace, carriage return, and line feed functions work as expected (even on inverted text), making it easy to add or edit titles and labels. A summary of escape sequences and keyboard operations affecting Graphics Text Mode is given in table 10-11.

Table 10-11. Graphics Text Functions

Key	Description
ESC t * d S	Selects the graphics image memory as the destination for all text. Characters entering from the keyboard or datacomm, are drawn as vectors in the graphics memory using the current text size and angle (see below). The graphics cursor indicates the position of the next character. Moving the graphics cursor will cause the next text line to begin at the new cursor position. The carriage return, line feed and backspace functions work normally.
ESC t * d T	Terminates Text Mode.
ESC t * m <size> M	Increases the character size from 1 to 8X. The smallest character is a 5 by 7 matrix in a 7 by 10 cell. Increasing the size makes the dots bigger while the character is still drawn as a 5 by 7 matrix.
ESC t * m <orientation> N	Sets the character orientation (multiples of 90 degrees).
ESC t * m O ESC t * m P	Turns slant on or off.
TAB	Spaces one graphics text character to the right. (The actual direction of movement will depend on the text orientation.)
CNTL K	(Vertical Tab). Spaces one graphics text line up. (The actual direction of movement will depend on the text orientation.)
In addition, the following keys function in the same manner as for alphanumeric text characters: RETURN , LINE FEED , BACK SPACE , ← , → , ↑ , ↓	

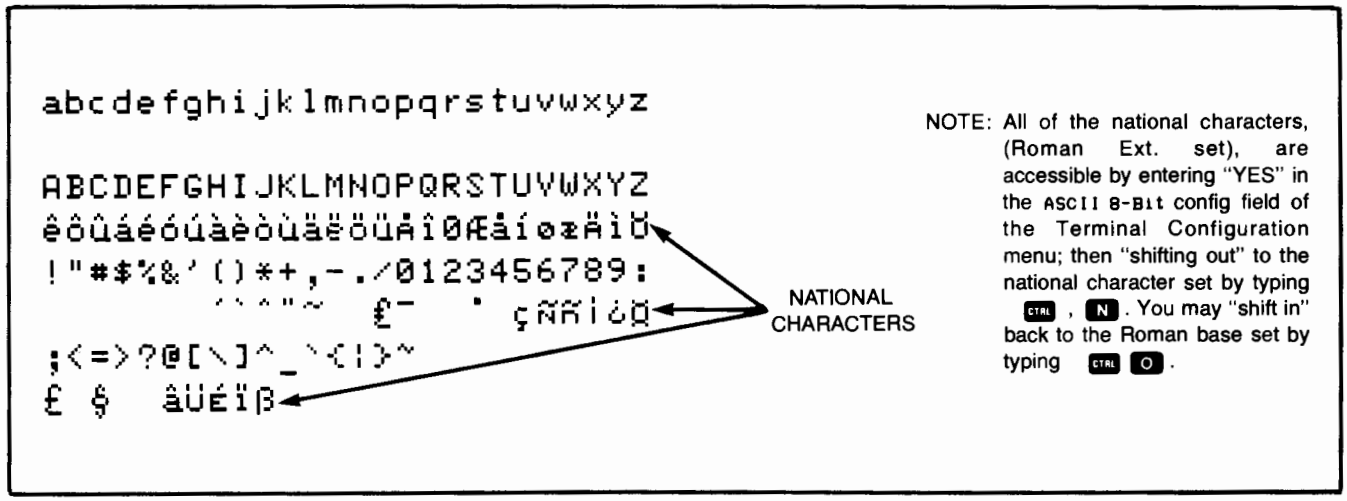


Figure 10-10. Graphics Text Characters

Program Control of Graphics Text

All of the parameters for graphics text can be set programmatically. Commands are of the form: ESC * m <parameter> <command>. The command can be alone or part of another ESC * m sequence.

SIZE. The ASCII characters 1 through 8 specify the character size for graphics text. A "1" indicates the smallest character, a 5 by 7 dot matrix character in a 7 by 10 cell. Increasing the size increases the size of the dots. If a text size of 1 is specified, each dot in the cell is one dot on the screen. A size of 2 uses 4 screen dots for each character dot (2 X 2), and so on (see figure 10-11). A size of "1" is the default.

Set Graphics Text Size: ESC * m <size> m

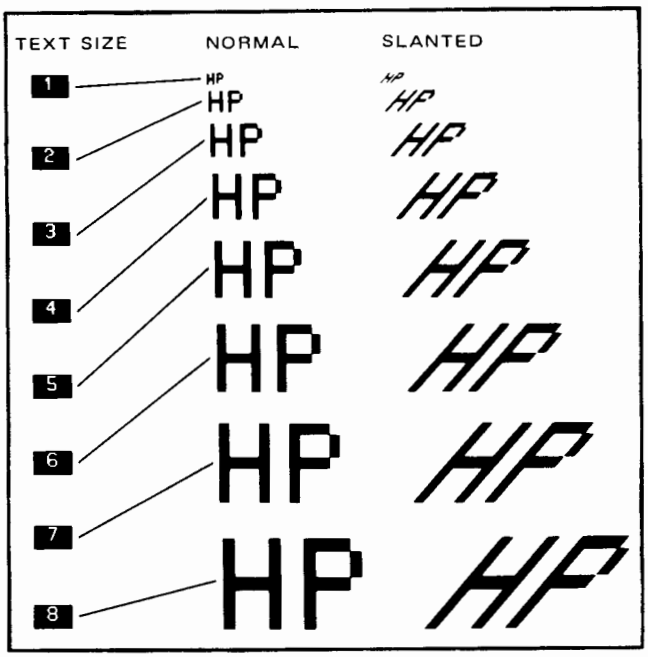


Figure 10-11. Graphics Text Sizes

TEXT DIRECTION. This command uses the ASCII characters 1 through 4 to specify the text orientation (see figure 10-12). This also changes the direction of line feed, carriage return, and backspace.

1 — Normal (upright, the default)
2 — Rotated 90 degrees counter clockwise
3 — Rotated 180 degrees counterclockwise (inverted)
4 — Rotated 270 degrees counter clockwise

Set Graphics Text Orientation: ESC * m <orientation> n

SLANT. The graphics text characters can be slanted 45 degrees for an italics effect.

Turn On Graphics Text Slant: ESC * m o

Turn Off Graphics Text Slant: ESC * m p

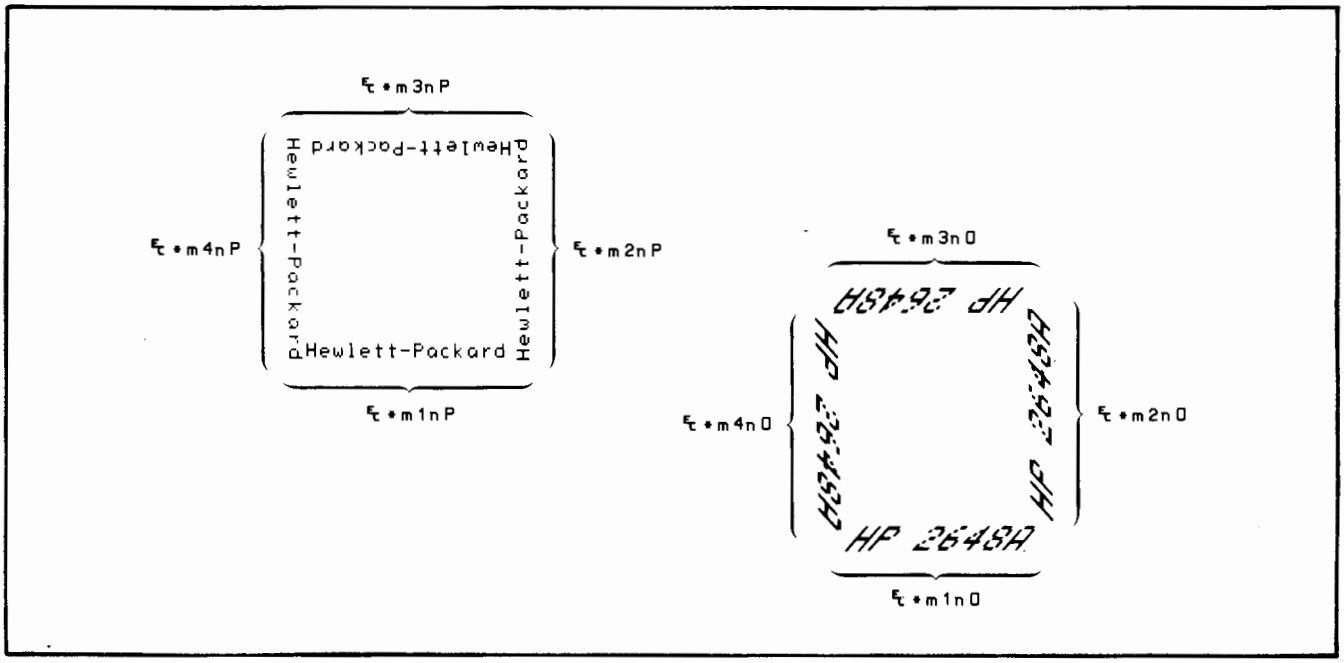
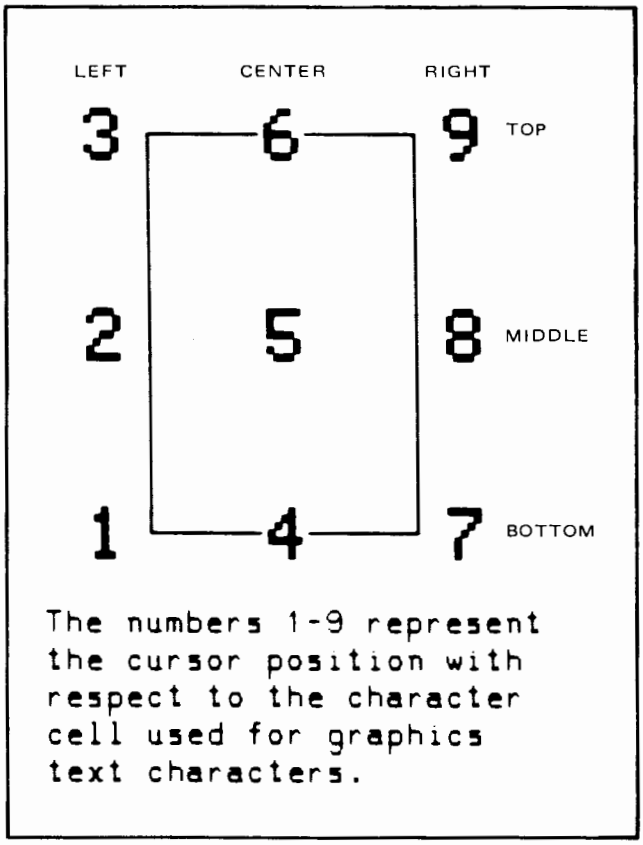


Figure 10-12. Graphics Text Direction

JUSTIFICATION/ORIGIN. Text strings can be automatically right or left justified, or centered about a specified point. An ASCII character 1 through 9 indicates the origin (justification and base line) for characters with respect to the current pen position. This function is useful when drawing labels. (Refer to the Label command.)

Set Graphics Text Justification: `ESC * m <origin> q`

If text is left justified, the current pen position is the left margin. Center causes the label to be centered on the pen position. Right justify selects the pen position as the right margin. Bottom, middle, and top select the base line for the line of text.



For example, if text was to be right justified and set with a base line on top of the normal character position, the number "9" would be used. Figure 10-13 illustrates the various text positions.

When centering or right justification is used, the text strings are buffered (stored) until all of the characters in the string have been received. The string end is detected by a CR or LF. The string is not displayed until the CR or LF is received. This may be confusing when entering text from the keyboard. The maximum length of a string when center or right justifying is 73 characters (not including the CR(LF)). In all cases, data written beyond the edge of the screen is lost. There is no automatic RETURN when the screen boundary is reached.

TURNING GRAPHICS TEXT ON AND OFF. Graphics text mode can be turned on or off from a program. These two commands use the `ESC * d` sequence but are discussed here under graphics text for completeness.

On. This command will cause Graphics Text Mode to be turned on. All displayable characters will be stored in the graphics memory. The drawing mode is initially set to jam mode to permit overstrike replacement of characters. A different mode, such as set or complement, can be selected at any time.

Text is drawn using the current text assignments for size and orientation. Graphics text mode accepts CR, LF, BS, HT, and VT as control characters. The ←, →, ↑, and ↓ keys can be used to position the graphics cursor in character increments.

Turn On Graphics Text Mode: `ESC * d S`

Turn Off Graphics Text Mode: `ESC * d T`

If the graphics cursor is moved, the graphics text margin is moved to the new cursor or pen position.

Characters are drawn using the current drawing mode (set, clear, or jam). If set mode is used, entering a character, backspacing, and entering a second character causes an overstrike. If jam mode is used, the new character will replace the old character.

If a lower case "s" is used, additional escape parameters can be appended to the sequence. Otherwise the next characters will be routed to the graphics memory.

Examples:

```
ESC * d s k 100,100 o B   set graphics memory
                          position cursor at 100,100
                          turn on cursor

ESC * d S  This is a text string%
```

Off. This sequence turns off graphics text mode and restores normal alphanumeric operation.

Turn Off Graphics Text Mode: `ESC * d t`

GRAPHICS TEXT STATUS. You can check the current text settings with a graphics text status request. Refer to Section VII, Status, for additional information.

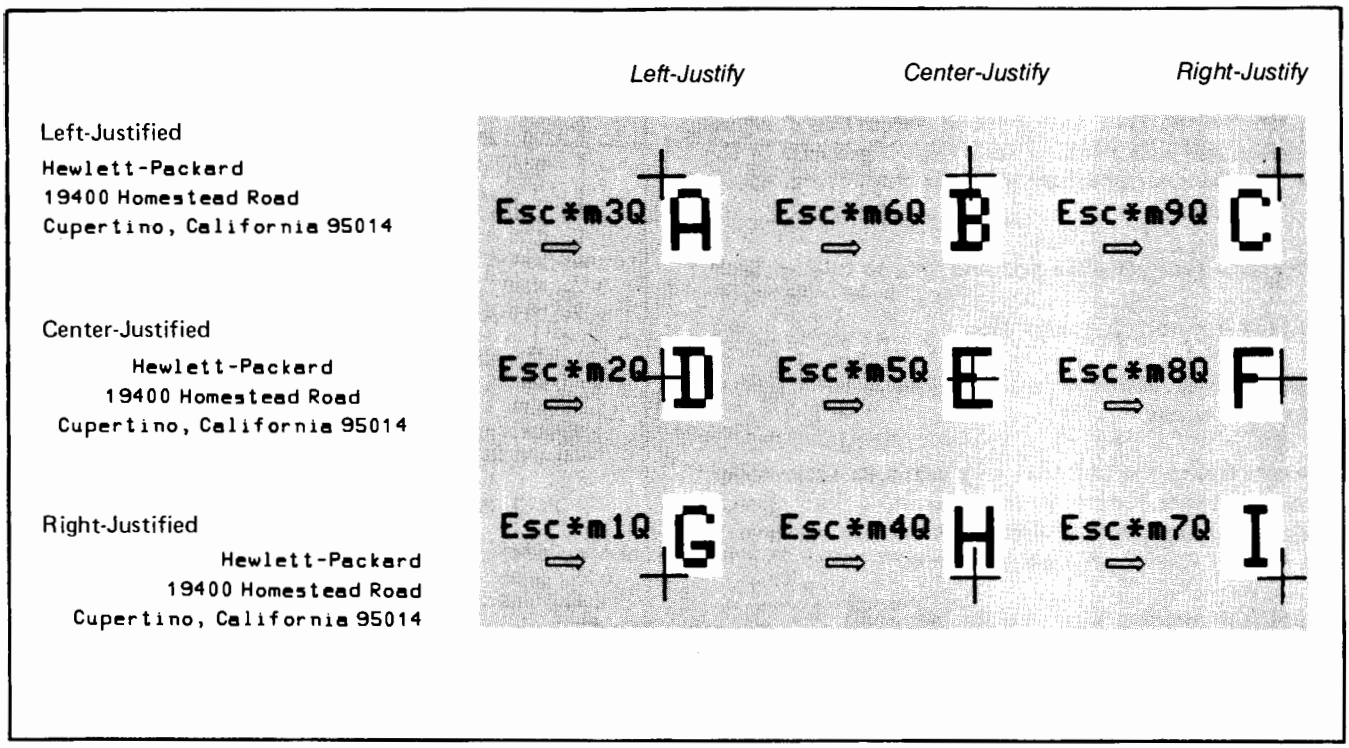


Figure 10-13. Graphics Text Justification

LABEL. This sequence is used to send a single record of graphics text to the terminal. The characters are stored in the graphics memory using the current text size, angle, slant, and justification. The label is drawn beginning at the current pen position.

Graphics Text Label: Esc * l <text string> %(r)

The record must end with a CR, LF, or both. A CR moves the pen to its original position when the label command was first received. An LF moves the pen down one line (character spacing). Note that the actual directions moved following a CR or LF depend on the text orientation selected.

The maximum record length is 73 characters, not including the Esc * l preamble or the CR(LF).

Example: Esc*lThis is a sample label%r

COMPATIBILITY MODE

Compatibility Mode allows the terminal to plot data intended for a terminal using a display with 1024 by 1024 addressable points. This mode makes it possible to use graphics programs developed for use with other graphics terminals with a minimum of reprogramming.

The terminal operates in two submodes while in Compatibility Mode. In Alphanumeric mode the terminal simply displays alphanumeric data on the screen as in normal operation. In Graphics mode the terminal responds to alphanumeric data as vector coordinates. Normally the terminal will be switched between these modes to display messages, plot graphics figures, and then display additional messages. These modes are controlled with several control sequences. (These sequences are ignored or acted on differently if the terminal is not set for Compatibility Mode.) Table 10-12 lists the terminal's responses to Compatibility Mode control sequences.

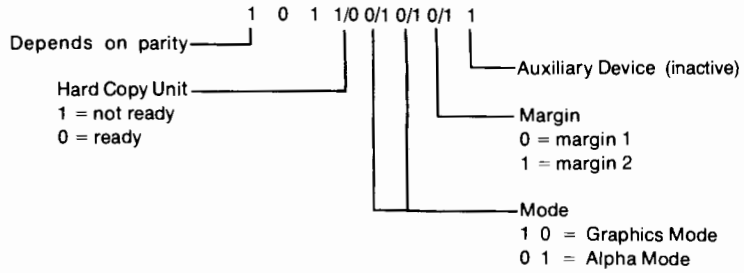
If delays are required, the baud rate can be lowered or fill characters added to prevent data loss when operating the terminal at high speeds. Refer to Section VI, Data Communications.

Vectors are drawn using the current line type and line drawing mode. This gives you the capability of drawing dotted and dashed lines, etc. by changing the program to send the additional escape sequences. In general, all of the normal features of the terminal (display enhancements, tape control, etc.) are available only in the Alphanumeric mode.

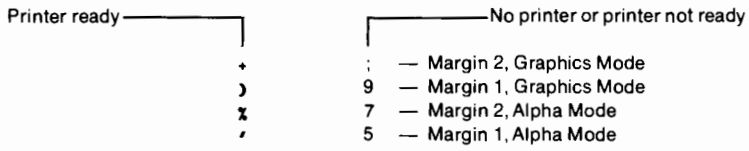
Compatibility Mode is turned on by selecting either scaled or unscaled operation. Escape sequences controlling Compatibility Mode begin with Esc*t. This preamble is then followed with one or more commands. These commands are listed in table 10-13. As in all other escape sequences, a captial letter ends the sequence. Figure 10-14 contains examples of typical escape sequences.

Table 10-12. Compatibility Mode Control Sequences

CONTROL SEQUENCE	DESCRIPTION	RESPONSE
^E ^E	Read status and alpha cursor position	<status byte><HI X><LO X> <HI Y><LO Y><terminator>



The terminal will return one of the following characters as the status byte:



^E ^B (20 ms delay) ^E ^D	Read graphics cursor position	<HI X><LO X><HI Y><LO Y><terminator>
^E ^B	Read graphics cursor position when key struck	<KEY><HI X><LO X><HI Y><LO Y><terminator>
^E ^H	Make hardcopy	
^E ^L	End graphics mode, clear screen, and home cursor	
^]	Go into graphics mode (draw vectors)	
^_	Go into alpha mode	
^H	Backspace (H'). Moves 1 space left (14 units)	
^I	Horizontal Tab (I'). Moves 1 space right (14 units)	
^X	End graphics mode	
^J	Line Feed (J'). Moves 1 line down (22 units)	
^K	Vertical Tab (K'). Moves 1 line up (22 units)	

NOTES

The terminal will normally respond with an ^K character when an ^D character is received. Compatibility Mode disables the terminal's ^D/^K handshake. Compatibility Mode causes most control codes to be ignored.

The Read Status, alpha cursor position, and graphic cursor position cause block transfers to the computer system. If the computer system does not use the DC1/DC2/DC1 handshake, `InhHndShk(G)` and `InhDC2(H)` in the Terminal Configuration menu must be "`YES`" for these transfers to occur. (Refer to "Terminal Configuration menu" in Section II.)

Compatibility Mode Configuration

Compatibility Mode operation is controlled by **Compat(P,Q)** field in the Terminal Configuration menu. This field can also be set programmatically using the "Esc & s . . ." sequence shown in table 10-13. The **P** and **Q** straps determine the terminal's mode of operation after being initialized (power up or full reset). The straps are interpreted as follows:

STRAPS		DESCRIPTION
P	Q	
0	0	Normal graphics operation
0	1	Unscaled Compatibility Mode
1	0	Scaled Compatibility Mode
1	1	Normal graphics operation

In addition, when in Compatibility Mode, you can select the following optional capabilities:

GRAPHIC INPUT TERMINATOR. You can select the terminator sent by the terminal following the input of cursor address information. The terminator can be a CR, CR and EOT, or no terminator.

PAGE FULL BUSY. When this strap is in, the keyboard will be locked after the 35th line of text is received from the computer. The terminal can be cleared by pressing the [GRAPH CLEAR] key. This strap is ignored in Unscaled Mode.

PAGE FULL BREAK. When this strap is in, the terminal will send a 200ms break signal to the computer after the 35th line of text is displayed. The terminal may also be set to BUSY (see Page Full Busy). When out, the strap will cause the cursor to home and the next 35 lines of text to be set with a left margin at x = 259. This strap is ignored in Unscaled Mode.

The commands to control these strap options are listed in table 10-13. Refer to the manual for the replaced graphics terminal for additional information on the operation of these straps and how they should be set.

Graphic Data

There are differences in display size (512 × 390 for the HP 2623A versus 1024 × 781 for other terminals) and line length (24 lines of 80 characters for the HP 2623A versus 35 lines of 74 characters for other terminals). See figure 10-15.

Table 10-13. Commands for Selecting Compatibility Mode

COMMAND	CODE
TURN SCALED COMPATIBILITY MODE ON (P open)	Esc & s 1 p 0 Q
TURN UNSCALED COMPATIBILITY MODE ON (Q open)	Esc & s 0 p 1 Q
TURN COMPATIBILITY MODE OFF (P,Q closed)	Esc & s 0 p 0 Q
The following commands simulate straps used on other graphics terminals:	
SET GRAPHICS INPUT TERMINATOR STRAP 0 — Carriage return only (Normal position) 1 — Carriage return and EOT 2 — No carriage return, no EOT	Esc * t <byte1> a
SET PAGE FULL BREAK STRAP 0 — Out (Normal position) 1 — In	Esc * t <byte1> b
SET PAGE FULL BUSY STRAP 0 — Out (Normal position) 1 — In	Esc * t <byte1> c
NOP	z

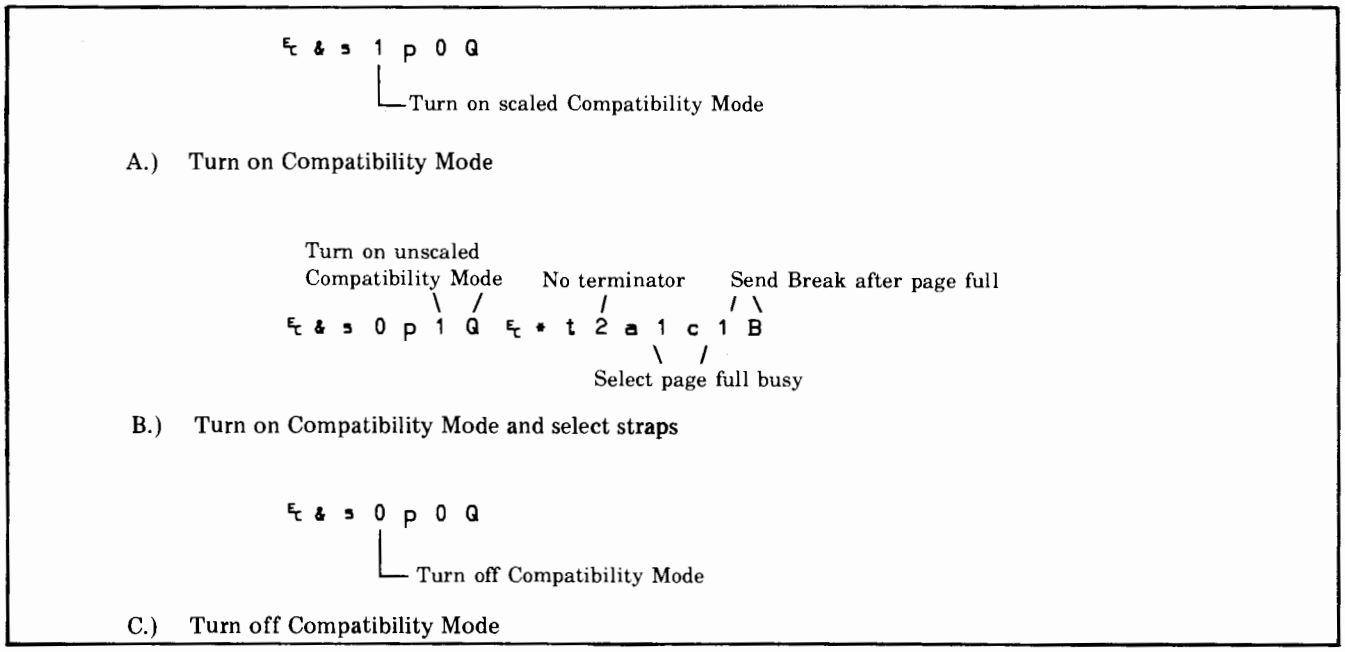


Figure 10-14. Turning on Compatibility Mode

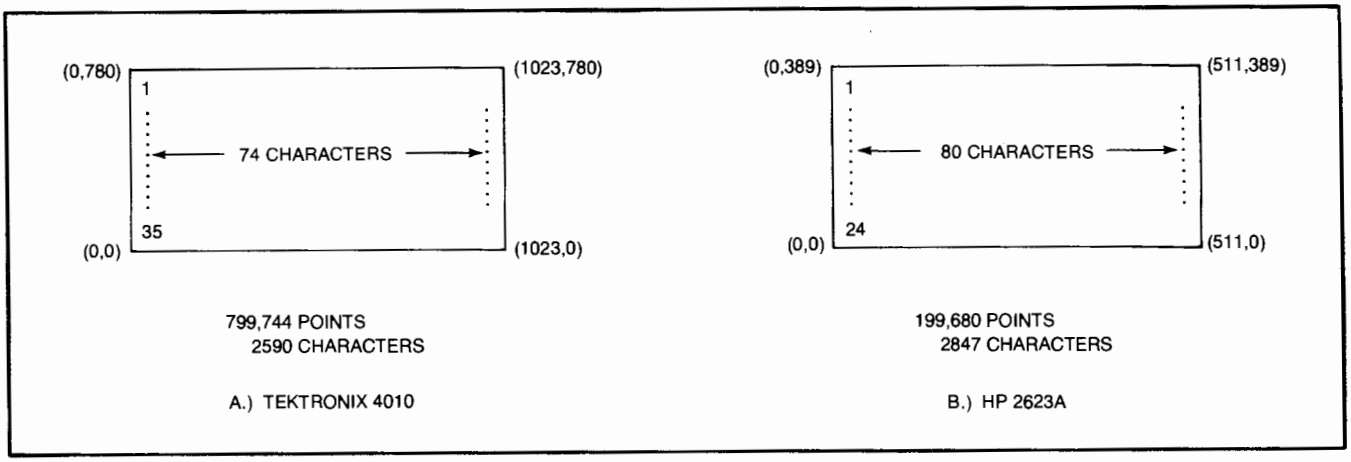


Figure 10-15. Comparison of a Terminal with 1024 × 780 Display and the HP 2623A

Graphic data can be drawn either scaled or unscaled. Scaling divides X coordinates by 2, and Y coordinates by 128/59. This maps the 1024 × 780 display into 512 by 390. This allows a program written for the 1024 × 780 terminal to run unchanged, and still display the entire picture (with some loss in resolution). The image doesn't cover the entire screen (only going to X = 512). The remainder can be used as a dialog area for alphanumeric text (see figure 10-16).

Unscaled mode shows a 512 by 390 subset of the 1024 × 780 picture. The area this covers can be changed by modifying the value of the relocatable origin (and redrawing the picture). The relocatable origin is subtracted from all incoming coordinates in unscaled mode. If this is set to 0,0 (the default) the range X = 0 to 511, Y = 0 to 389 will be displayed (see figure 10-17a).

Setting the origin to 0,360 would cover the X = 0 to 511, Y = 360 to 749. To display an area larger than 512 × 390, you must change the scaling statements in the program. The advantage of unscaled mode over scaled mode is that unscaled allows you to use the entire available display area.

Graphics Data Format

In Compatibility mode the graphics data is formatted as two-byte coordinate values. The lower five bits of each byte are used to make a 10 bit (0-1023) coordinate. Data sent to the terminal must have the "Y" coordinate sent first; <Upper Y> <Lower Y> <Upper X> <Lower X>.

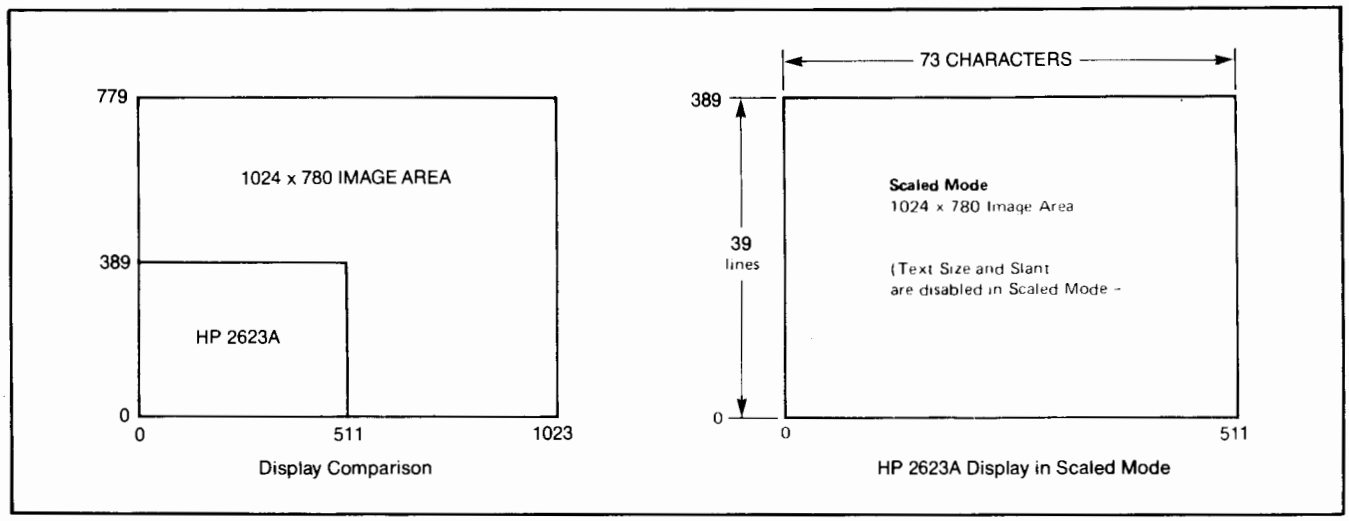


Figure 10-16. Scaled Data

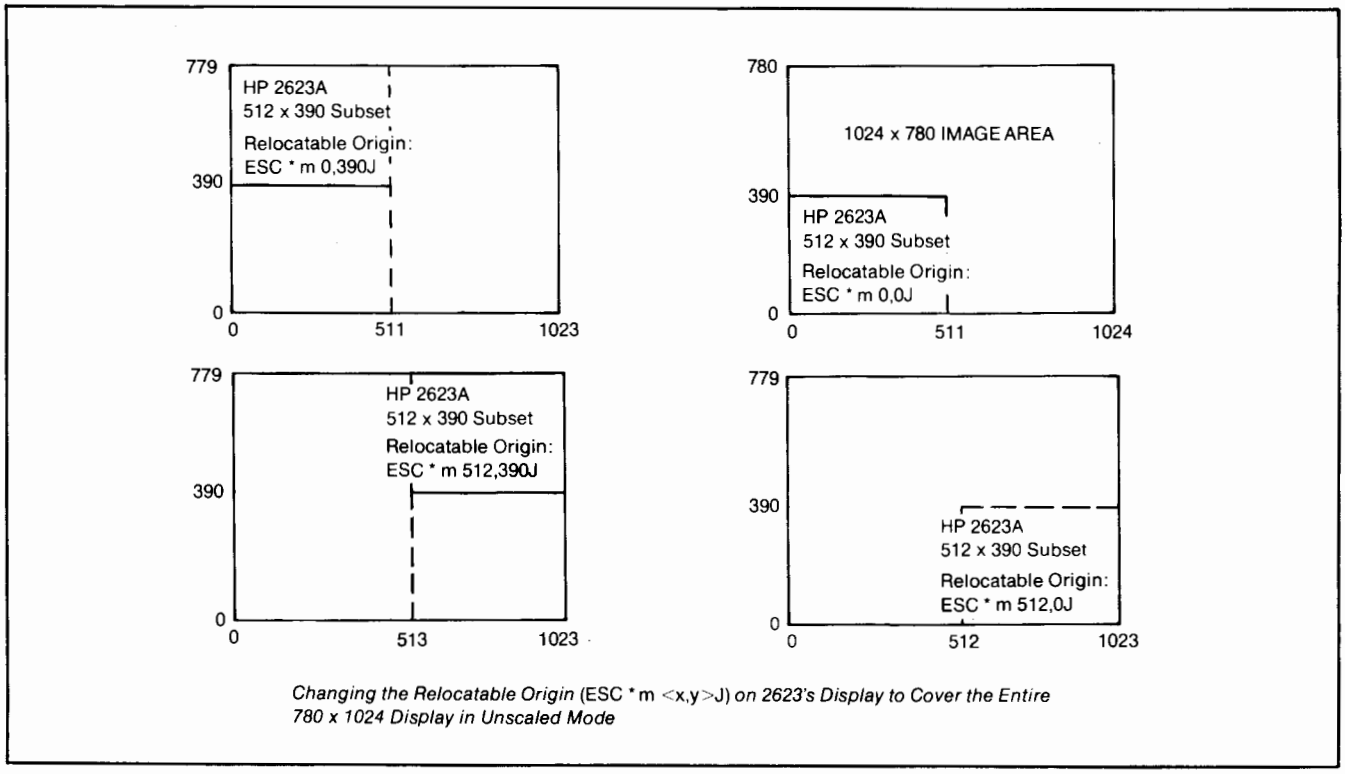


Figure 10-17. Unscaled Data

When data is returned to the computer (cursor position, etc.), the X coordinate is returned first; <Upper X> <Lower X> <Upper Y> <Lower Y>.

Data bytes sent to the terminal use bits 6 and 7 to indicate the byte is an Upper byte, a lower Y, or a lower X. Bit 8 (parity) is not used.

Bits		
7	6	
0	1	Upper X or Y byte
1	0	Lower X byte
1	1	Lower Y byte

These identifying bits allow you to send only the changed portion of a four byte address. The following data bytes must always be sent:

- Lower X byte
- Any changed byte
- Lower Y byte if the Upper X byte has changed

Table 10-14 can be used to determine address bytes. For example, to plot the points A (0,0), B (0,31), C (256,31), D (256,0) the following sequence would be used:

Text

Text can be placed in either the alphanumeric memory or in the graphics memory. If the terminal is set for alphanumeric text, the text will be sent to the alphanumeric memory. This is generally the most useful, as text can be scrolled, edited, erased, etc. without affecting the graphics image. If you select graphics text (ESC * d 3), text will go into the graphics memory. Text to be written to the graphics memory can be scaled or rotated. (Refer to Graphics Text in this section for additional information.)

When text is written to the graphics memory, the graphics cursor is moved to indicate where the next character will be stored. (The alphanumeric cursor is only used when data is stored in the alphanumeric memory.) This differs from terminals that have only one mode for text and display the graphics cursor only when waiting for graphic input from the user.

SCALED MODE GRAPHICS TEXT. In Scaled Mode, text is initially written into the graphics memory, the size is fixed to allow for 35 lines of text. The text angle is set at 0 degrees and unslanted. The text origin is set to the left and bottom. These settings allow the "Page Full" feature to work properly and existing software to run without changes. If you do not require the Page Full feature, you can not change the text settings. You can redirect the text to the alphanumeric memory.

UNSCALED MODE GRAPHICS TEXT. In Unscaled Mode, the text size is unchanged and graphics text mode is not initially turned on. Text is stored in the alpha numeric memory unless the graphics text mode is specifically enabled.

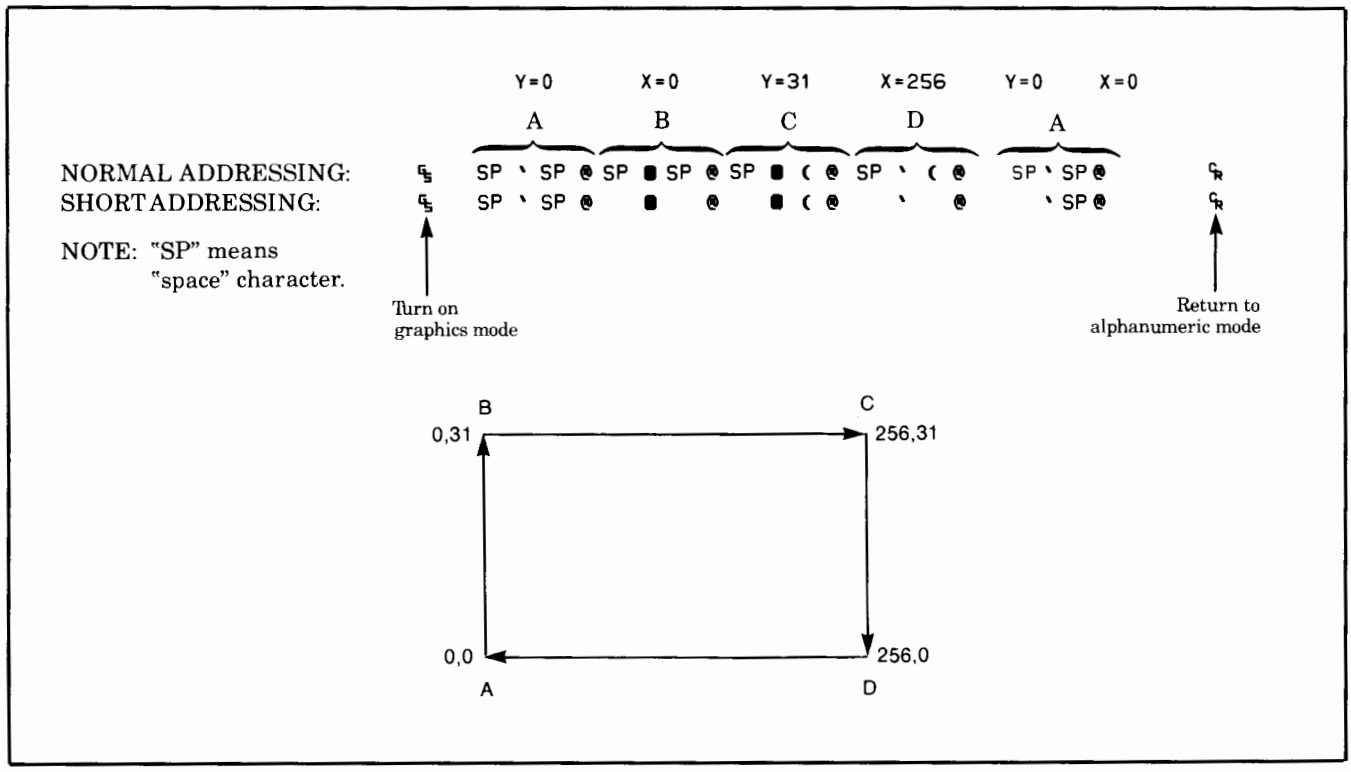


Table 10-14. Coding of Compatibility Mode Graphics Data

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
0	32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	96	`	64	@
1	33	65	97	129	161	193	225	257	289	321	353	385	417	449	481	97	a	65	A
2	34	66	98	130	162	194	226	258	290	322	354	386	418	450	482	98	b	66	B
3	35	67	99	131	163	195	227	259	291	323	355	387	419	451	483	99	c	67	C
4	36	68	100	132	164	196	228	260	292	324	356	388	420	452	484	100	d	68	D
5	37	69	101	133	165	197	229	261	293	325	357	389	421	453	485	101	e	69	E
6	38	70	102	134	166	198	230	262	294	326	358	390	422	454	486	102	f	70	F
7	39	71	103	135	167	199	231	263	295	327	359	391	423	455	487	103	g	71	G
8	40	72	104	136	168	200	232	264	296	328	360	392	424	456	488	104	h	72	H
9	41	73	105	137	169	201	233	265	297	329	361	393	425	457	489	105	i	73	I
10	42	74	106	138	170	202	234	266	298	330	362	394	426	458	490	106	j	74	J
11	43	75	107	139	171	203	235	267	299	331	363	395	427	459	491	107	k	75	K
12	44	76	108	140	172	204	236	268	300	332	364	396	428	460	492	108	l	76	L
13	45	77	109	141	173	205	237	269	301	333	365	397	429	461	493	109	m	77	M
14	46	78	110	142	174	206	238	270	302	334	366	398	430	462	494	110	n	78	N
15	47	79	111	143	175	207	239	271	303	335	367	399	431	463	495	111	o	79	O
16	48	80	112	144	176	208	240	272	304	336	368	400	432	464	496	112	p	80	P
17	49	81	113	145	177	209	241	273	305	337	369	401	433	465	497	113	q	81	Q
18	50	82	114	146	178	210	242	274	306	338	370	402	434	466	498	114	r	82	R
19	51	83	115	147	179	211	243	275	307	339	371	403	435	467	499	115	s	83	S
20	52	84	116	148	180	212	244	276	308	340	372	404	436	468	500	116	t	84	T
21	53	85	117	149	181	21	245	277	309	341	373	405	437	469	501	117	u	85	U
22	54	86	118	150	182	214	246	278	310	342	374	406	438	470	502	118	v	86	V
23	55	87	119	151	183	215	247	279	311	343	375	407	439	471	503	119	w	87	W
24	56	88	120	152	184	216	248	280	312	344	376	408	440	472	504	120	x	88	X
25	57	89	121	153	185	217	249	281	313	345	377	409	441	473	505	121	y	89	Y
26	58	90	122	154	186	218	250	282	314	346	378	410	442	474	506	122	z	90	Z
27	59	91	123	155	187	219	251	283	315	347	379	411	443	475	507	123	{	91	[
28	60	92	124	156	188	220	252	284	316	348	380	412	444	476	508	124	|	92	\
29	61	93	125	157	189	221	253	285	317	349	381	413	445	477	509	125	}	93	]
30	62	94	126	158	190	222	254	286	318	350	382	414	446	478	510	126	~	94	^
31	63	95	127	159	191	223	255	287	319	351	383	415	447	479	511	127	■	95	_
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	← DEC.			
SP	!	"	#	$	%	&	'	(	)	*	+	,	−	.	/	← ASCII			
High Order X & Y																			

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
512	544	576	608	640	672	704	736	768	800	832	864	896	928	960	992	96	`	64	@
513	545	577	609	641	673	705	737	769	801	833	865	897	929	961	993	97	a	65	A
514	546	578	610	642	674	706	738	770	802	834	866	898	930	962	954	98	b	66	B
515	547	579	611	643	675	707	739	771	803	835	867	899	931	963	995	99	c	67	C
516	548	580	612	644	676	708	740	772	804	836	868	900	932	964	996	100	d	68	D
517	549	581	613	645	677	709	741	773	805	837	869	901	933	965	997	101	e	69	E
518	550	582	614	646	678	710	742	774	806	838	870	902	934	966	998	102	f	70	F
519	551	583	615	647	679	711	743	775	807	839	871	903	935	967	999	103	g	71	G
520	552	584	616	648	680	712	744	776	808	840	872	904	936	968	1000	104	h	72	H
521	553	585	617	649	681	713	745	777	809	841	873	905	937	969	1001	105	i	73	I
522	554	586	618	650	682	714	746	778	810	842	874	906	938	970	1002	106	j	74	J
523	555	587	619	651	683	715	747	779	811	843	875	907	939	971	1003	107	k	75	K
524	556	588	620	652	684	716	748	780	812	844	876	908	940	972	1004	108	l	76	L
525	557	589	621	653	685	717	749	781	813	845	877	909	941	973	1005	109	m	77	M
526	558	590	622	654	686	718	750	782	814	846	878	910	942	974	1006	110	n	78	N
527	559	591	623	655	687	719	751	783	815	847	879	911	943	975	1007	111	o	79	O
528	560	592	624	656	688	720	752	784	816	848	880	912	944	976	1008	112	p	80	P
529	561	593	625	657	689	721	753	785	817	849	881	913	945	977	1009	113	q	81	Q
530	562	594	626	658	690	722	754	786	818	850	882	914	946	978	1010	114	r	82	R
531	563	595	627	659	691	723	755	787	819	851	883	915	947	979	1011	115	s	83	S
532	564	596	628	660	692	724	756	788	820	852	884	916	948	980	1012	116	t	84	T
533	565	597	629	661	693	725	757	789	821	853	885	917	949	981	1013	117	u	85	U
534	566	598	630	662	694	726	758	790	822	854	886	918	950	982	1014	118	v	86	V
535	567	599	631	663	695	727	759	791	823	855	887	919	951	983	1015	119	w	87	W
536	568	600	632	664	696	728	760	792	824	856	888	920	952	984	1016	120	x	88	X
537	569	601	633	665	697	729	761	793	825	857	889	921	953	985	1017	121	y	89	Y
538	570	602	634	666	698	730	762	794	826	858	890	922	954	986	1018	122	z	90	Z
539	571	603	635	667	699	731	763	795	827	859	891	923	955	987	1019	123	{	91	[
540	572	604	636	668	700	732	764	796	828	860	892	924	956	988	1020	124	|	92	\
541	573	605	637	669	701	733	765	797	829	861	893	925	957	989	1021	125	}	93	]
542	574	606	638	670	702	734	766	798	830	862	894	926	958	990	1022	126	~	94	^
543	575	607	639	671	703	735	767	799	831	863	895	927	959	991	1023	127	■	95	_
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	← DEC			
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	← ASCII			
High Order X & Y																			

Example: 340Y,70X is found as follows:

340Y = 42 (upper Y) 116 (Lower Y) 70X = 34 (Upper X) 70 (Lower X)

```
340Y,70X → • t " F
```

Escape Codes

APPENDIX A

KEY(S)			CODE	FUNCTION
TERMINAL CONTROL FUNCTION				
[ENTER]	(as used in Local mode)		↯ 0	Copy memory to destination(s)
[TABS]	margins / tabs / col	SET TAB	↯ 1	Set tab
[TABS]	margins tabs / col	CLEAR TAB	↯ 2	Clear tab
[TABS]	margins / tabs / col	CLR ALL TABS	↯ 3	Clear all tabs
[TABS]	margins / tabs / col	LEFT MARGIN	↯ 4	Set left margin
[TABS]	margins / tabs / col	RIGHT MARGIN	↯ 5	Set right margin
[TABS]	margins / tabs / col	CLR ALL MARGINS	↯ 9	Clear all margins
			↯ *	Delay one second
[↑]			↯ A	Cursor up
[↓]			↯ B	Cursor down
[→]			↯ C	Cursor right
[←]			↯ D	Cursor left

KEY(S)		CODE	FUNCTION
[SHIFT] [CTRL] [RESET]		↯ E	Hard reset (power on reset)
[HOME] [↑]		↯ F	Cursor home down
[RETURN]	(with Auto LF disabled)	↯ G	Move cursor to left margin
[↖]		↯ H	Cursor home up
[TAB]	or [BACK TAB]	↯ I	Horizontal tab
[CLEAR DISPLAY]		↯ J	Clear display from cursor to end of memory
[CLEAR LINE]		↯ K	Clear line from cursor to end of line
[INSERT LINE]		↯ L	Insert line
[DELETE LINE]		↯ M	Delete line
[DEL CHAR]		↯ P	Delete character
[INS CHAR]		↯ Q	Start insert character mode
[INS CHAR]		↯ R	End insert character (↯ Q)
[ROLL UP]		↯ S	Roll up
[ROLL DOWN]		↯ T	Roll down
[NEXT PAGE]		↯ U	Next page
[PREV PAGE]		↯ V	Previous page

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)		
	↑ W	Format mode on
	↑ X	Format mode off
, DISPLAY FUNCTION	↑ Y	Display Functions mode on
, DISPLAY FUNCTNS *	↑ Z	Display Functions mode off
	↑ [	Start unprotected field
	↑]	End unprotected/transmit-only field
	↑ ^	Primary terminal status request
	↑ '	Sense cursor position (relative)
	↑ a	Sense cursor position (absolute)
	↑ b	Unlock keyboard
	↑ c	Lock keyboard
	↑ d	Transmit a block of text to computer
	↑ f	Modem disconnect
	↑ g	Soft reset
	↑ h	Cursor home up
	↑ i	Backtab
	↑ j	Begin User Key Definition mode
or or	↑ k	End User Keys Definition mode
MEMORY LOCK	↑ l	Begin Memory Lock mode
MEMORY LOCK *	↑ M	End Memory Lock mode
	↑ p	Default definition for user definable function key f1
	↑ q	Default definition for user definable function key f2
	↑ r	Default definition for user definable function key f3
	↑ s	Default definition for user definable function key f4
	↑ t	Default definition for user definable function key f5
	↑ u	Default definition for user definable function key f6
	↑ v	Default definition for user definable function key f7
	↑ w	Default definition for user definable function key f8

KEY(S)		CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)			
service keys	TERMINAL TEST	ESC z	Initiate terminal self test
TERMINAL TEST			
		ESC ~	Secondary terminal status request
Shift Num		ESC &k <x>0	x = 0 Enable numeric pad x = 1 Enable graphics pad

CURSOR CONTROL OPERATIONS	
NOTE Columns and rows are numbered starting with 0 as the leftmost column and the top row.	
ESC &a <col>c <rows>Y	Moves the cursor to column "col" and screen row "row" or the screen (screen relative addressing).
ESC &a <col>c <row>R	Moves the cursor to column "col" and row "row" in memory (absolute addressing).
ESC &a ± <col>c ± <row>Y	Moves the cursor to column "col" and row "row" (on the screen) relative to its present position ("col" and "row" are signed integers). A positive number indicates right or downward movement and a negative number indicates left or upward movement.
ESC &a ± <col>c ± <row>R	Moves the cursor to column "col" and row "row" relative to its present cursor position in memory ("col" and "row" are signed integers). A positive number indicates right or downward movement and a negative number indicates left or upward movement.

CONFIGURATION OPERATIONS

↑ &q 0L	Unlock configuration.
↑ &q 1L	Lock configuration.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	X
↑ &k <x>A	AUTO LF	OFF	x=0
		ON	x=1
↑ &k <x>B	BLOCK	OFF	x=0
		ON	x=1
↑ &k <x>C	Caps Lock	OFF	x=0
		ON	x=1
↑ &k <x>I	ASCII8Bits	NO	x=0
		YES	x=1
↑ &k <x>J	FrameRate	60	x=0
		50	x=1
↑ &k <x>L	LocalEcho	OFF	x=0
		ON	x=1
↑ &k <x>M	MODIFY ALL	OFF	x=0
		ON	x=1
↑ &k <x>N	SPOW Latch	OFF	x=0
		ON	x=1
↑ &k <x>P	Caps Mode	OFF	x=0
		ON	x=1
↑ &k <x>R	REMOTE	OFF	x=0
		ON	x=1
↑ &k <x>S	Print Type for Integral Printer	Normal	x=0
		Expanded	x=1
		Compressed	x=2
↑ &s <x>A	xmitFnctn(A)	NO	x=0
		YES	x=1
↑ &s <x>B	SPOW(B)	NO	x=0
		YES	x=1
↑ &s <x>C	InhEolWrp(C)	NO	x=0
		YES	x=1
↑ &s <x>D	Line/Page (D)	LINE	x=0
		PAGE	x=1
↑ &s <x>G	IndHndShk(G)	NO	x=0
		YES	x=1
↑ &s <x>H	Inh DC2(H)	NO	x=0
		YES	x=1
HP 2623A only			
↑ &s <x>N	Esc Xfer(N)	NO	x=0
		YES	x=1
↑ &s <x>p<y>Q	Compat(P,Q)	UNSCALED	x=0,y=1
		SCALED	x=1,y=0
		OFF	x=1,y=1
			or
			x=0,y=L

DATA OPERATIONS

The following escape sequences control data transfer to and from the integral and external printers and display memory. Only effective if printer present. The HP 2622A does NOT support an external printer.

ESC &p <a>d <b>d <c>d <Y>

Copies "Y" amount of data to destination devices "a", "b", and "c". As many destinations as desired can be specified.

a, b, and c	DEVICE
3	Display.
4	External printer
6	Integral printer.

y	ACTION
B	Copy the Line in which the cursor is located.
F	Copy the display screen from the line in which the cursor is located (cursor line) to the last displayed line.
M	Copy the contents of display memory from the cursor line to the end of memory.

ESC &p <x>^

Requests the status of device "x".

x	DEVICE
4	External printer
6	Integral printer.

ESC &k <x>S

Enables Expanded, Compressed, or Normal Character mode for the integral printer as designated by the character "X".

x	ACTION
0	Disable both Expanded and Compressed Character modes.
1	Initiate Expanded Character mode.
2	Initiate Compressed Character mode.

ESC &p <x>p <y>u <z>C

Performs the action specified by "z" on device "y".

z	ACTION
0	Generates 1 form feed.
1	Space "x" lines.
2-10	Generates 1 form feed.
11	Turn on Log Bottom mode.
12	Turn on Log Top mode.
13	Turn off any logging mode.
14	Print normal characters.
15	Print expanded characters.
16	Print compressed characters.
17	Turn on Report mode.
18	Turn on Metric Report mode.
19	Turn off any Report mode.
20	Turn on Record Mode

y	DEVICE
4	Internal or external printer depending on the PRINTER-CODE4 entry on the External Device Configuration menu.
6	Integral printer.
3	Display.

<x>p

x Is an optional parameter which can define the character to turn off Record mode. Or it may be used in conjunction with 1C to specify the number of lines to advance.

DATA OPERATIONS (continued)

`↖&p <a>d <b>d <c>d <x>W <data string>`	Transfers "x" bytes of the data string from the computer to the selected destination device in binary form.
`↖&p <a>d <b>d <c>d W <data string>`	Transfers the data string, in ASCII form, from the computer to the printer selected as the destination device. The string is terminated either by the 256th byte or by an ASCII line feed character. **If "d" (destination) parameter is not specified, the selected "to" devices in the softkey tree are assumed to be the destinations.**

FORMAT MODE

`↖[`	Starts a field.
`↖]`	Ends the field.

FUNCTION KEY AND ERROR MESSAGE OPERATIONS

To enable and disable the function keys (F1 thru F8), use the following escape sequence:

`↖&j <x>`

x	MEANING
A	Display the Modes set of function key labels.
B	Enable the User function keys. (The user key labels are displayed.)
C	**Clear message and restore the current key labels.**
@	Remove the function key labels from the screen. The User function keys, however, are still active.

To enable or disable the Function Control keys:

S	Disables the [RESET], [MODES], and [USER KEYS] keys.
R	Enables the [RESET], [MODES], and [USER KEYS] keys.

To define the function keys:

`↖&f <attribute>a <key>k <label length>d <string length>L <label> <string>`

TERM	SYMBOL	MEANING	DEFAULT
Attribute (a)	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
Key (k)	1	[f1] function key	
	2	[f2] function key	
	3	[f3] function key	
	4	[f4] function key	
	5	[f5] function key	
	6	[f6] function key	
	7	[f7] function key	
	8	[f8] function key	
Label length (d)	0 thru 16	Number of characters in the label.	0
String length (l)	0 thru 80	Number of characters in the string.	1
	−1	Clears the content of the string.	
Label	(none)	The label is entered at this point in the sequence.	
String	(none)	The character string is entered at this point in the sequence.	

To replace the function key definition with your own message:

`↖&j <string length>L <message>`

"String Length"—A number (up to 160) indicating the number of characters in the string.

"Message"—The content of the message.

DISPLAY ENHANCEMENTS OPERATIONS

To start and end display enhancements:

ESC & d <char> Selects the display enhancement indicated by "char" to begin at the present cursor position.

	"char"														
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright									X	X	X	X	X	X	X	X
Under-line					X	X	X	X					X	X	X	X
Inverse Video			X	X			X	X			X	X			X	X
Blinking		X		X		X		X		X		X		X		X
End Enhance-ment	X															

ALTERNATE CHARACTER SET SELECTION

Selects alternate character set as:

ESC) 0	Base Set
ESC) A	Base Set
ESC) B	Line drawing
ESC) C	Blank character set

ESC & w 12F	Turns on the display window (top 24 rows)
ESC & w 13F	Turns off the display window

KEY(S)	CODE	FUNCTION

GRAPHICS CONTROL SEQUENCES

ESC • <control sequence>

d	= Display control
l	= Graphics text label
m	= Mode control
p	= Plot control
s	= Status
t	= Compatibility mode

DISPLAY CONTROL

ESC •d <parameters>

GRAPH CLEAR	a	Clear graphics memory
— — —	b	Set graphics memory
GRAPH DSPLY	c	Turn on graphics display
GRAPH DSPLY	d	Turn off graphics display
ALPHA DSPLY	e	Turn on alphanumeric display
ALPHA DSPLY	f	Turn off alphanumeric display
GRAPH CURSOR	k	Turn on graphics cursor
GRAPH CURSOR	l	Turn off graphics cursor

KEY(S)	CODE	FUNCTION

DISPLAY CONTROL (Continued)

— — —	<x,y>	o	Move graphics cursor absolute
< , ^ , > , v	<x,y>	p	Move graphics cursor incremental
— — —		q	Turn on alphanumeric cursor
— — —		r	Turn off alphanumeric cursor
— — —		s	Turn on graphics text mode
— — —		t	Turn off graphics text mode
— — —		z	NOP

Example: Clear the graphics display, position the cursor at x=100, y=100, and turn the cursor on.

```
ESC • d a 100,100o K
```

GRAPHICS LABEL

ESC •l <text label><CR>, <LF>, <ETX>, or <EOT>

Example: Send the text "X=TIME, Y=TEMP"

```
ESC • l X=TIME, Y=TEMP <LF>
```

VECTOR DRAWING MODE

`ESC * m <parameters>`

KEY(S)		CODE	FUNCTION
— — —	<mode>	a	Select drawing mode (0–4)*
— — —	<line type>	b	Select line type (1–11)**
— — —	<pattern scale>	c	Define line pattern (2 bytes)
— — —	<pattern>	d	Define area shading pattern (8 bytes)
— — —	<x1,y1,x2,y2>	e	Fill area, absolute
— — —	<x1,y1,x2,y2>	f	Fill area, relocatable
— — —	<x,y>	j	Select relocatable origin
— — —		k	Set relocatable origin to current pen position
— — —		l	Set relocatable origin to graphics cursor position
— — —	<size>	m	Set graphics text size (1–8) (1 default)
— — —	<rotation>	n	Set graphics text orientation (1–4) (1 default)
— — —		o	Turn on text slant
— — —		p	Turn off text slant
— — —	<0–9>	q	Set graphics text origin
— — —		r	Set graphics defaults
— — —		z	NOP

* 0 (no effect), 1 (clear), 2 (set), 3 (complement), 4 (jam)

** 1 (solid line), 2 (user line pattern), 3 (user area pattern), 4 (line #1), 5 (line #2), 6 (line #3), 7 (line #4), 8 (line #5), 9 (line #6), 10 (line #7), 11 (point plot)

Example: Select the set drawing mode, a graphics text size of 2 and slanted. Set the text to be center justified.

```
ESC * m 2a 2m o 4Q
```

The text is not displayed until a CR is executed.

Graphics text must be turned on.

```
ESC *dS.
```

PLOTTING COMMANDS

`ESC * p <parameters>`

KEY(S)		CODE	FUNCTION
— — —		a	Lift the pen
— — —		b	Lower the pen
— — —		c	Use graphics cursor as new point
— — —		d	Draw a point at the current pen position and lift the pen
— — —		e	Set relocatable origin to the current pen position
— — —		f	Data is ASCII absolute
— — —		g	Data is ASCII incremental
— — —		h	Data is ASCII relocatable
— — —		i	Data is absolute
— — —		j	Data is short incremental
— — —		k	Data is incremental
— — —		l	Data is relocatable
— — —		z	NOP

Example: Draw a box 25 units wide and 10 units high, beginning at x=100, y=50.

```
ESC * p a f 100 50 g 25,0 0,10 -25,0 0, -10Z
```

KEY(S)	CODE	FUNCTION
GRAPHICS STATUS		
	ESC * s <parameter> ^	
— — —	1	Read device I.D.
— — —	2	Read pen position
— — —	3	Read graphics cursor position
— — —	4	Read cursor position and wait for key
— — —	5	Read display size
— — —	6	Read graphics capabilities
— — —	7	Read graphics text status
— — —	8	Read zoom status
— — —	9	Read relocatable origin
— — —	10	Read reset status
— — —	11	Read area shading
— — —	12	Read dynamics
Example: Read text status. ESC * s 7 ^		

KEY(S)	CODE	FUNCTION
COMPATIBILITY MODE		
	ESC * t <parameter>	
— — —	<0/1/2> a	Set graphics input terminator (0=CR, 1=CR EOT, 2=none)
— — —	<0/1> b	Set Page Full Break strap (0=out, 1=in)
— — —	<0/1> c	Set Page Full Busy strap (0=out, 1=in)
— — —	z	NOP
Example: Select a CR input terminator and set the Page Full Busy strap. ESC * t 0a 1C		

Keyboards and Character Sets

APPENDIX B

NATIONAL KEYBOARDS

Figures B-1 through B-7 show the various national keyboards which are available as options 001 through 006. Note that these options also include the extended character set ROMs which support all of the national languages, and the line drawing set.

If you order the standard USASCII keyboard and you wish the terminal to include the extended character set ROMs, then you must specifically order the ROMs as option 202.

The French keyboard (option 003), when delivered, is physically arranged in the AZERTY layout; a keycap extraction tool comes with it. To change the keyboard to the QWERTY layout, you must physically rearrange the A, Z, Q, and W keys as shown in figure B-4.

7-BIT VS. 8-BIT OPERATION

You configure the terminal for 7-bit operation by setting the `ASCII 8 Bits` field of the Terminal Configuration menu to "`NO`".

When the terminal is configured for 7-bit operation, the ASCII `<SO>` code (which enables the active alternate character set) applies through the end of the current line; when the cursor moves to the next lower line you must once again issue a `<SO>` if you wish to continue typing in the active alternate character set.

You configure the terminal for 8-bit operation by setting the `ASCII 8 Bits` field of the applicable Terminal Configuration menu to "`YES`".

When the terminal is configured for 8-bit operation, the ASCII `<SO>` code applies until the next subsequent `<SI>` code (which disables the active alternate character set), even if the `<SI>` occurs several lines below the `<SO>` code.

ISO/ASCII CHARACTER SET

Table B-1 shows the standard ISO/ASCII character set.

If the terminal includes the extended character set ROMs and is configured for 7-bit operation, the shaded characters in table B-1 are replaced on the screen with the following characters (depending upon which national language is specified in the Terminal Configuration menu):

LANGUAGE	KEYBOARD OPTION #	DECIMAL VALUE 35	64	91	92	93	94	96	123	124	125	126
USASCII	(standard)	#	@	[	\	]	^	`	{	¦	}	~
Swedish/Finnish	001	#	É	Ä	Ö	Å	ü	é	ä	ö	å	ü
Danish/Norwegian	002	#	@	Æ	Ø	Å	^	`	æ	ø	å	~
French	003	£	à	°	ç	§	^	`	é	ù	è	¨
German	004	£	§	Ä	Ö	ü	^	`	ä	ö	ü	ß
United Kingdom	005	£	@	[	\	]	^	`	{	¦	}	~
Spanish	006	#	@	¡	Ñ	¿	°	`	{	ñ	}	~

If the terminal is configured for a foreign language but does NOT include the extended character set ROMs, the characters in the above table are displayed on the screen as spaces.

If the terminal is configured for 8-bit operation and a foreign language, the active alternate character is set to "Roman Ext".

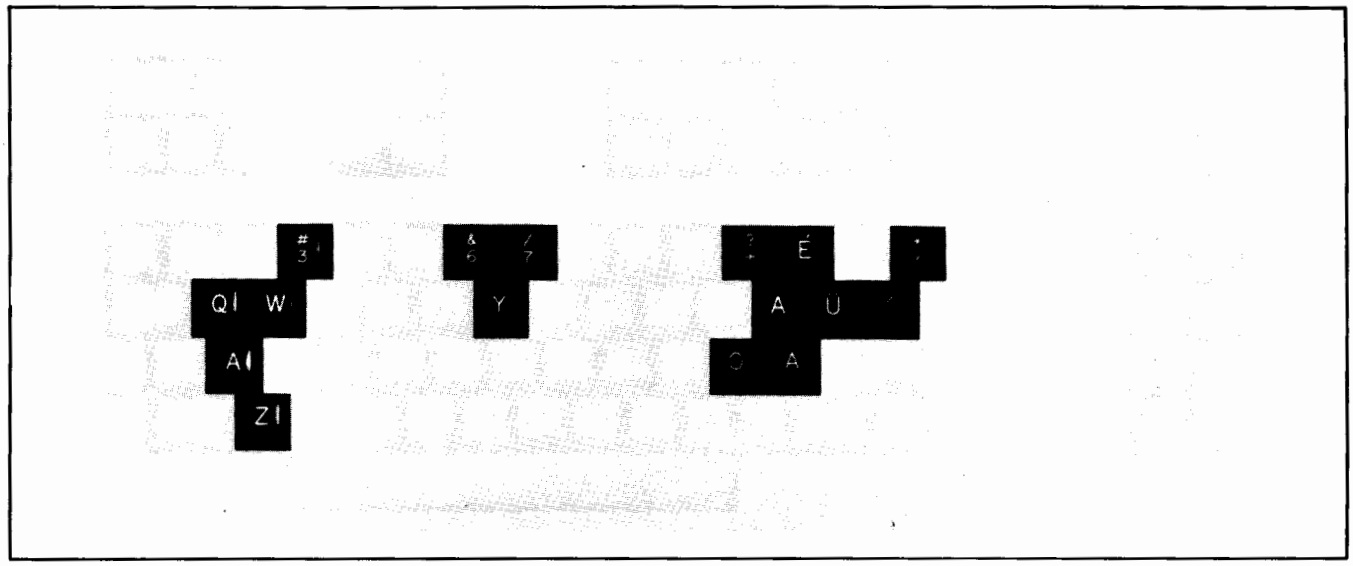


Figure B-1. Swedish/Finnish Keyboard (Option 001)

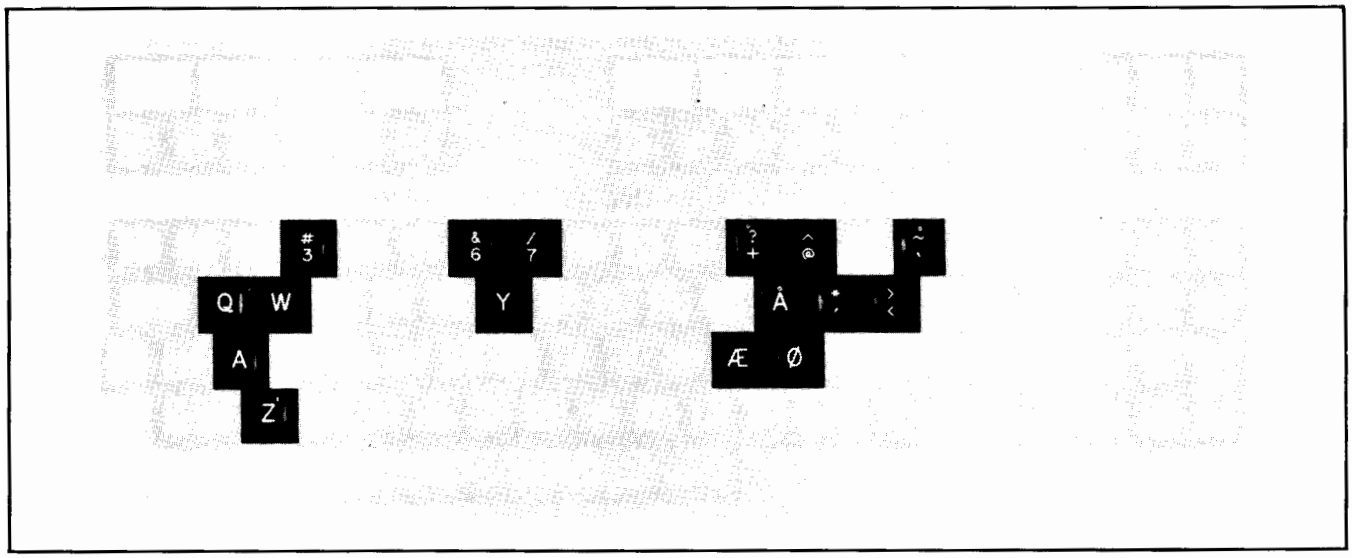


Figure B-2. Danish/Norwegian Keyboard (Option 002)

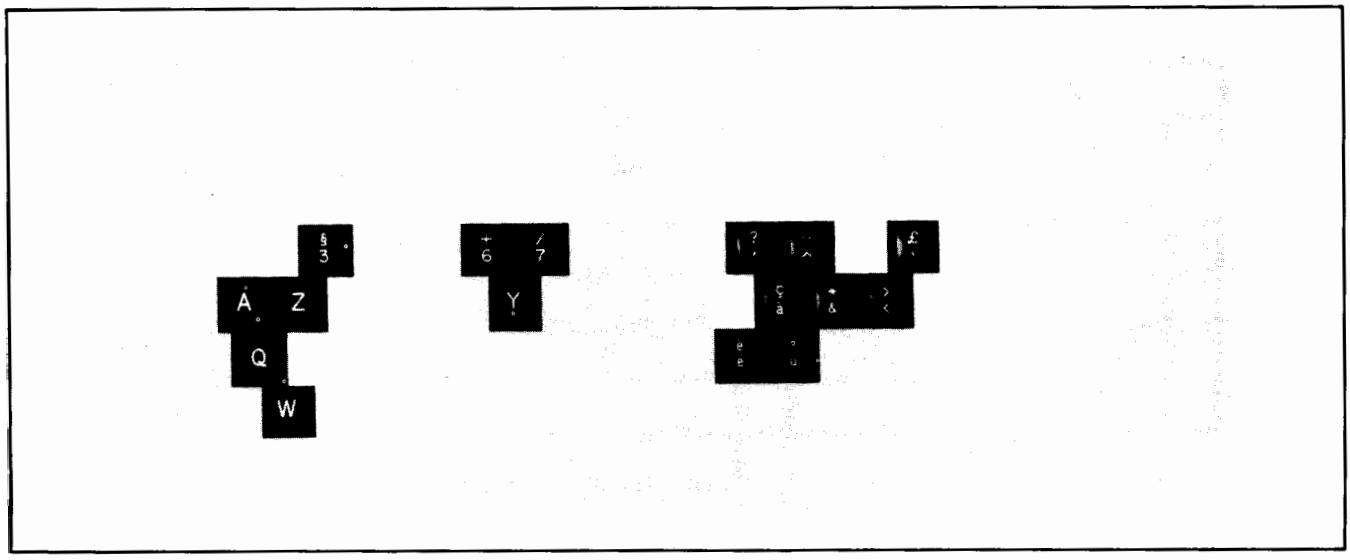


Figure B-3. French Keyboard (Option 003), AZERTY Layout

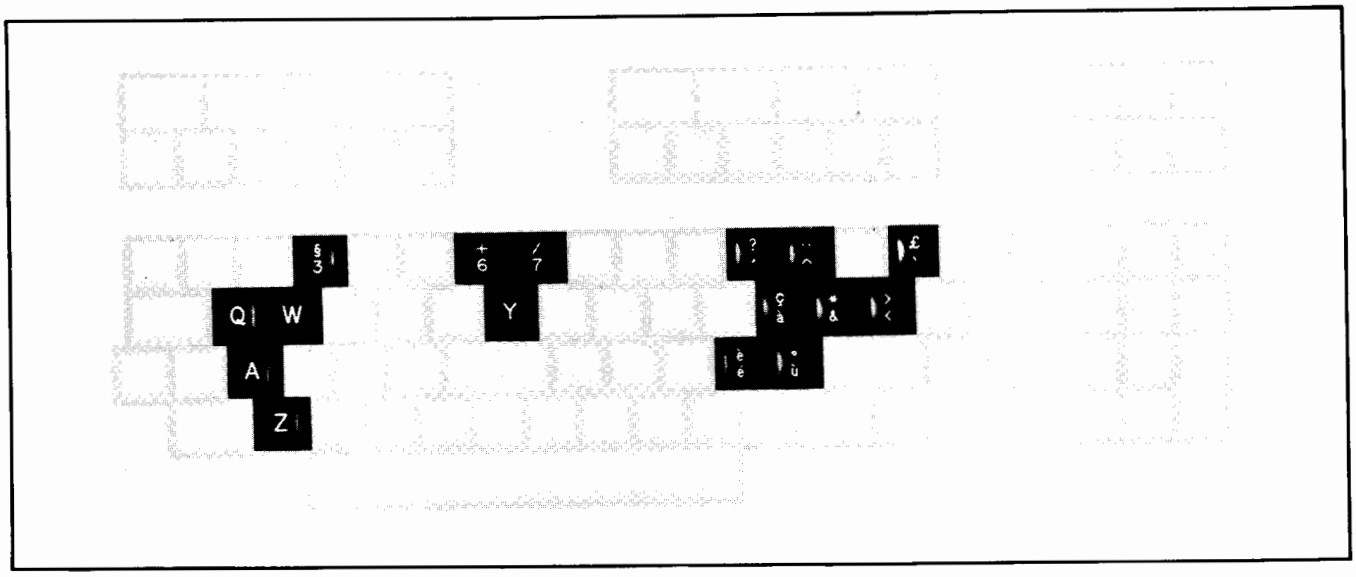


Figure B-4. French Keyboard (Option 003), QWERTY Layout

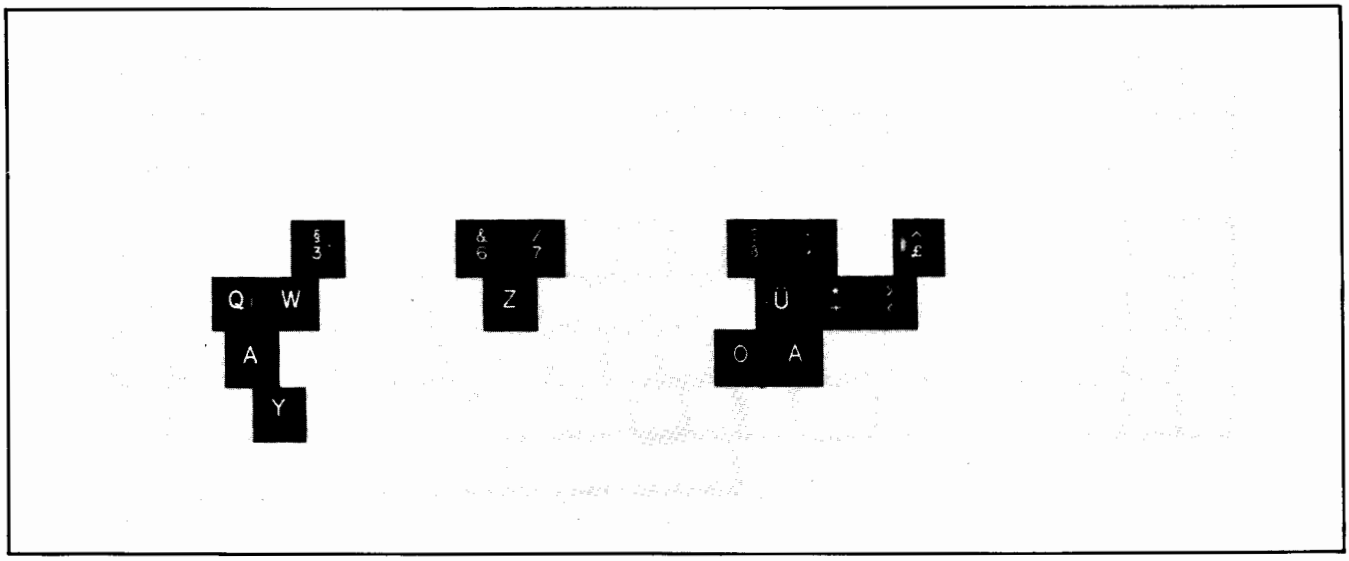


Figure B-5. German Keyboard (Option 004)

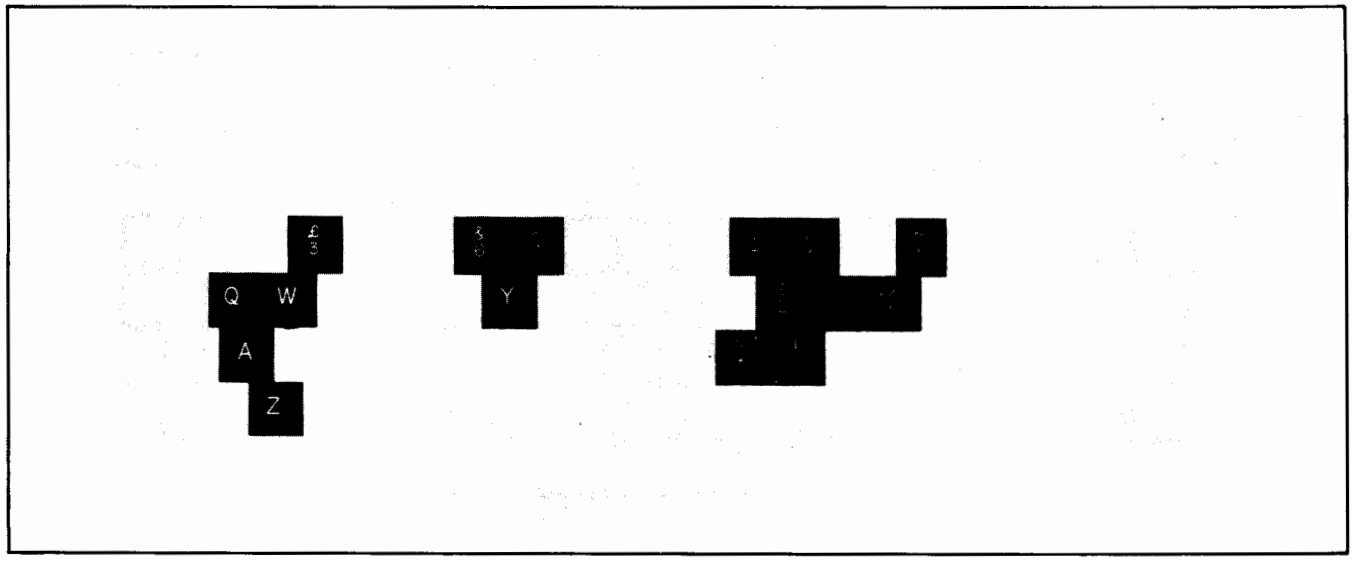


Figure B-6. United Kingdom Keyboard (Option 005)

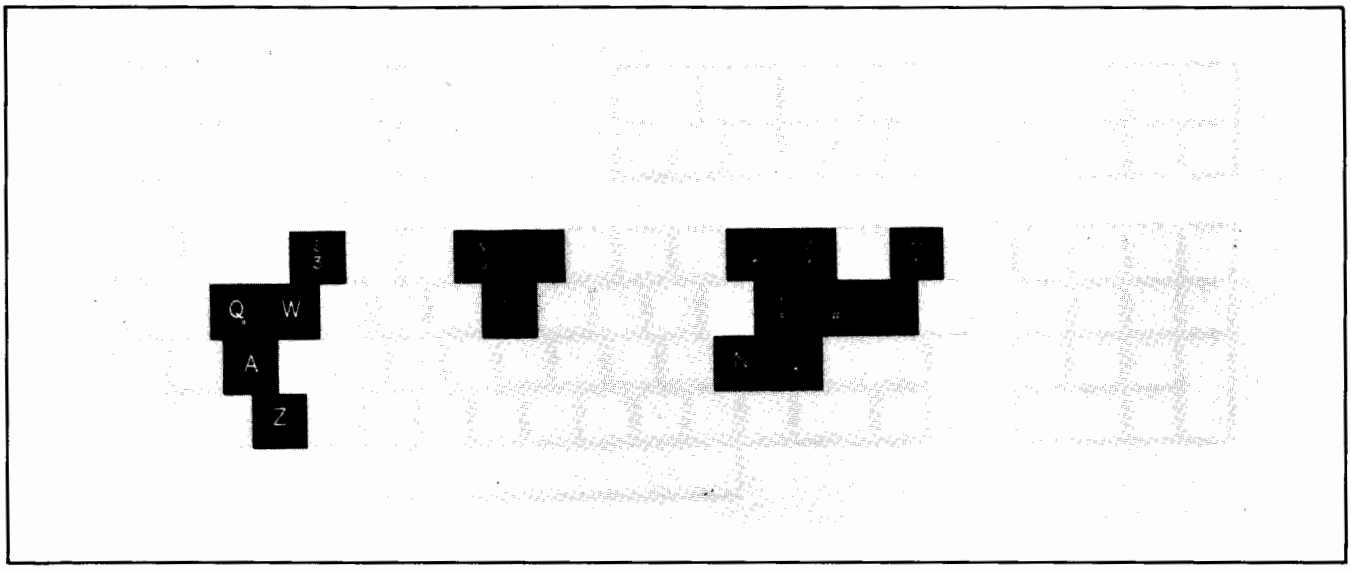


Figure B-7. Spanish Keyboard (Option 006)

EXTENDED ROMAN SET

If the terminal is configured for 8-bit operation and "Roman Ext" is selected as the active alternate character set, the entire character set comprising tables B-1 and B-2 is used when interpreting character codes. In such a case, the eighth data bit determines which table applies. If bit 8 is a zero, the character code is interpreted according to table B-1. If bit 8 is a one, the character code is interpreted according to table B-2.

Note that if the terminal does NOT include the extended character set ROMs, the character codes are still interpreted as described above but those codes which map to table B-2 are displayed on the screen as spaces.

As with any of the alternate character sets, you enable the Extended Roman set with a <SO> control code (control-N) and disable it with a <SI> control code (control-O).

The extended character set is used by the HP 300 and HP 250 computer systems and the HP 2631 and HP 2608 printers.

LINE DRAWING SET

If your terminal contains the line drawing character set option, you can construct forms and tables by combining different types of line segments. Each individual type of line seqment is associated with one of the alphanumeric or symbol keys as shown in figure B-8. Figure B-9 contains the keystrokes used for generating a sample data entry form.

Note that you use the "shift out" (SO) and "shift in" (SI) control codes to shift from the standard ASCII character set to the line drawing set and back again. For example:

1. To shift from ASCII to the line drawing set, use SO by pressing the N key while holding the CTRL key down.
2. To shift from the line drawing set back to ASCII, use SI by pressing the O key while holding the CTRL key down.

See Appendix C further for information on using the line drawing set.

ASCII/EBCDIC CHARACTER CODES

Table B-3 summarizes the entire 128-code ASCII character set, table B-4 presents the decimal, octal, and hexadecimal codes for the ASCII character set, and table B-5 presents the decimal, octal and hexadecimal codes for the EBCDIC character set.

Table B-1. Standard ISO/ASCII Character Codes

BIT 7 6 5 / 4 3 2 1	CONTROL (CNTL) CHARACTERS 0 0 0	0 0 1	DISPLAYABLE CHARACTERS 0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0000	@ 0 $^{N}_{U}$ NUL	P 16 $^{D}_{L}$ DLE	SP	0	@	P	`	p
0001	A 1 $^{S}_{H}$ SOH	Q 17 $^{D}_{1}$ DC1	!	1	A	Q	a	q
0010	B 2 $^{S}_{X}$ STX	R 18 $^{D}_{2}$ DC2	"	2	B	R	b	r
0011	C 3 $^{E}_{X}$ ETX	S 19 $^{D}_{3}$ DC3	#	3	C	S	c	s
0100	D 4 $^{E}_{T}$ EOT	T 20 $^{D}_{4}$ DC4	$	4	D	T	d	t
0101	E 5 $^{E}_{N}$ ENQ	U 21 $^{N}_{A}$ NAK	%	5	E	U	e	u
0110	F 6 $^{A}_{K}$ ACK	V 22 $^{S}_{Y}$ SYN	&	6	F	V	f	v
0111	G 7 BEL	W 23 $^{E}_{B}$ ETB	'	7	G	W	g	w
1000	H 8 $^{B}_{S}$ BS	X 24 $^{C}_{A}$ CAN	(	8	H	X	h	x
1001	I 9 $^{H}_{T}$ HT	Y 25 $^{E}_{M}$ EM	)	9	I	Y	i	y
1010	J 10 $^{L}_{F}$ LF	Z 26 $^{S}_{U}$ SUB	*	:	J	Z	j	z
1011	K 11 $^{V}_{T}$ VT	[27 $^{E}_{S}$ ESC	+	;	K	[	k	{
1100	L 12 $^{F}_{F}$ FF	\ 28 $^{F}_{S}$ FS	,	<	L	\	l	|
1101	M 13 $^{C}_{R}$ CR	] 29 $^{G}_{S}$ GS	-	=	M	]	m	}
1110	N 14 $^{S}_{O}$ SO	^ 30 $^{R}_{S}$ RS	.	>	N	^	n	~
1111	O 15 $^{S}_{I}$ SI	_ 31 $^{U}_{S}$ US	/	?	O	_	o	DEL

Table B-2. Extended Roman Character Codes

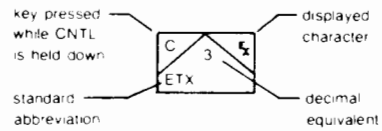
BIT 7 6 5 / 4 3 2 1	B_8 = 1 EXTENDED ROMAN CHARACTERS 0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0000				¯ / –	â	Â		
0001					ê	î		
0010					ô	Ø		
0011				•	û	fi		
0100					á	å		
0101				Ç	é	í		
0110				Ñ	ó	ø		
0111				ñ	ú	æ		
1000			‘	¡	à	Ä		
1001			’	¿	è	ì		
1010			^	¤	ò	Ö		
1011			¨	£	ù	ü		
1100			~		ä	ß		
1101				§	ë	ï		
1110					ö			
1111			£		ü			

$^{A}_{K}$ — ACKNOWLEDGE
$^{B}_{E}$ — BELL
$^{B}_{S}$ — BACKSPACE
$^{C}_{N}$ — CANCEL LINE
$^{C}_{R}$ — CARRIAGE RETURN
$^{D}_{L}$ — DATA LINK ESCAPE
$^{D}_{1}$ — DEVICE CONTROL 1
$^{D}_{2}$ — DEVICE CONTROL 2
$^{D}_{3}$ — DEVICE CONTROL 3
$^{D}_{4}$ — DEVICE CONTROL 4
DEL — DELETE
$^{E}_{M}$ — END OF MEDIUM
$^{E}_{N}$ — ENQUIRY
$^{E}_{T}$ — END OF TRANSMISSION

$^{E}_{T}$ — ESCAPE
$^{E}_{B}$ — END OF BLOCK
$^{E}_{X}$ — END OF TEXT
$^{F}_{F}$ — FORM FEED
$^{F}_{S}$ — FILE SEPARATOR
$^{G}_{S}$ — GROUP SEPARATOR
$^{H}_{T}$ — HORIZONTAL TAB
$^{L}_{F}$ — LINE FEED
$^{N}_{A}$ — NEGATIVE ACKNOWLEDGE
$^{R}_{S}$ — RECORD SEPARATOR
$^{S}_{I}$ — SHIFT IN
$^{S}_{O}$ — SHIFT OUT
SP — SPACE

$^{S}_{H}$ — START OF HEADING
$^{S}_{X}$ — START OF TEXT
$^{S}_{U}$ — SUBSTITUTE
$^{S}_{Y}$ — SYNCHRONOUS IDLE
$^{U}_{S}$ — UNIT SEPARATOR
$^{V}_{T}$ — VERTICAL TAB

Control Character Legend



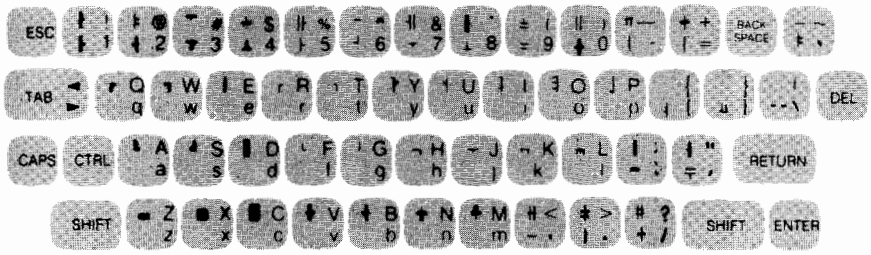


Figure B-8. Line Drawing Set Elements

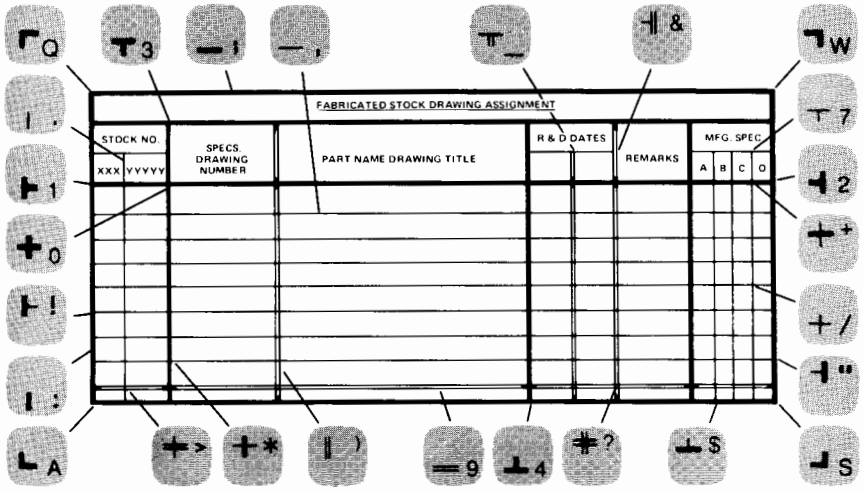


Figure B-9. Sample Data Entry Form

Table B-3. ASCII Character Set

DECIMAL VALUE	GRAPHIC	COMMENTS	ALTERNATE CHARACTER	DECIMAL VALUE	GRAPHIC	COMMENTS
0	^@	Null	@^c	64	@	Commercial at
1	^A	Start of heading	A^c	65	A	Uppercase A
2	^B	Start of text	B^c	66	B	Uppercase B
3	^C	End of text	C^c	67	C	Uppercase C
4	^D	End of transmission	D^c	68	D	Uppercase D
5	^E	Enquiry	E^c	69	E	Uppercase E
6	^F	Acknowledge	F^c	70	F	Uppercase F
7	^G	Bell	G^c	71	G	Uppercase G
8	^H	Backspace	H^c	72	H	Uppercase H
9	^I	Horizontal tabulation	I^c	73	I	Uppercase I
10	^J	Line feed	J^c	74	J	Uppercase J
11	^K	Vertical tabulation	K^c	75	K	Uppercase K
12	^L	Form feed	L^c	76	L	Uppercase L
13	^M	Carriage return	M^c	77	M	Uppercase M
14	^N	Shift out	N^c	78	N	Uppercase N
15	^O	Shift in	O^c	79	O	Uppercase O
16	^P	Data link escape	P^c	80	P	Uppercase P
17	^Q	Device control 1 (X-ON)	Q^c	81	Q	Uppercase Q
18	^R	Device control 2	R^c	82	R	Uppercase R
19	^S	Device control 3 (X-OFF)	S^c	83	S	Uppercase S
20	^T	Device control 4	T^c	84	T	Uppercase T
21	^U	Negative acknowledge	U^c	85	U	Uppercase U
22	^V	Synchronous idle	V^c	86	V	Uppercase V
23	^W	End of transmission block	W^c	87	W	Uppercase W
24	^X	Cancel	X^c	88	X	Uppercase X
25	^Y	End of medium	Y^c	89	Y	Uppercase Y
26	^Z	Substitute	Z^c	90	Z	Uppercase Z
27	^[	Escape	[^c	[1] 91	[	Opening bracket
28	^\	File separator	\^c	[2] 92	\	Reverse slant
29	^]	Group separator	]^c	[1] 93	]	Closing bracket
30	^^	Record separator	^^c	[1] 94	^	Circumflex
31	^_	Unit separator	_^c	[2] 95	_	Underscore
32		Space (Blank)		96	`	Grave accent
33	!	Exclamation point		97	a	Lowercase a
34	"	Quotation mark		98	b	Lowercase b
35	#	Number sign		99	c	Lowercase c
36	$	Dollar sign		100	d	Lowercase d
37	%	Percent sign		101	e	Lowercase e
38	&	Ampersand		102	f	Lowercase f
39	'	Apostrophe		103	g	Lowercase g
40	(	Opening parenthesis		104	h	Lowercase h
41	)	Closing parenthesis		105	i	Lowercase i
42	*	Asterisk		106	j	Lowercase j
43	+	Plus		107	k	Lowercase k
44	,	Comma		108	l	Lowercase l
45	-	Hyphen (Minus)		109	m	Lowercase m
46	.	Period (Decimal)		110	n	Lowercase n
47	/	Slant		111	o	Lowercase o
48	0	Zero		112	p	Lowercase p
49	1	One		113	q	Lowercase q
50	2	Two		114	r	Lowercase r
51	3	Three		115	s	Lowercase s
52	4	Four		116	t	Lowercase t
53	5	Five		117	u	Lowercase u
54	6	Six		118	v	Lowercase v
55	7	Seven		119	w	Lowercase w
56	8	Eight		120	x	Lowercase x
57	9	Nine		121	y	Lowercase y
58	:	Colon		122	z	Lowercase z
59	;	Semicolon		[2]123	{	Opening (left) brace
60	<	Less than		[2]124	|	Vertical line
61	=	Equals		[2]125	}	Closing (right) brace
62	>	Greater than		[2]126	~	Tilde
63	?	Question mark		127		Delete

Notes: 1. The equivalent EBCDIC character uses a different graphic.
2. No equivalent character exists in EBCDIC.

Table B-4. ASCII (7-Bit) Character Codes

GRAPHIC	DEC	OCT	HEX
NUL	0	0	0
SOH	1	1	1
STX	2	2	2
ETX	3	3	3
EOT	4	4	4
ENQ	5	5	5
ACK	6	6	6
BEL	7	7	7
BS	8	10	8
HT	9	11	9
LF	10	12	A
VT	11	13	B
FF	12	14	C
CR	13	15	D
SO	14	16	E
SI	15	17	F
DLE	16	20	10
DC1	17	21	11
DC2	18	22	12
DC3	19	23	13
DC4	20	24	14
NAK	21	25	15
SYN	22	26	16
ETB	23	27	17
CAN	24	30	18
EM	25	31	19
SUB	26	32	1A
ESC	27	33	1B
FS	28	34	1C
GS	29	35	1D
RS	30	36	1E
US	31	37	1F
SP	32	40	20
!	33	41	21
"	34	42	22
#	35	43	23
$	36	44	24
%	37	45	25
&	38	46	26
'	39	47	27
(	40	50	28
)	41	51	29
*	42	52	2A
+	43	53	2B
,	44	54	2C
-	45	55	2D
.	46	56	2E
/	47	57	2F
0	48	60	30
1	49	61	31
2	50	62	32
3	51	63	33
4	52	64	34
5	53	65	35
6	54	66	36
7	55	67	37
8	56	70	38
9	57	71	39
:	58	72	3A
;	59	73	3B
<	60	74	3C
=	61	75	3D
>	62	76	3E
?	63	77	3F

GRAPHIC	DEC	OCT	HEX
@	64	100	40
A	65	101	41
B	66	102	42
C	67	103	43
D	68	104	44
E	69	105	45
F	70	106	46
G	71	107	47
H	72	110	48
I	73	111	49
J	74	112	4A
K	75	113	4B
L	76	114	4C
M	77	115	4D
N	78	116	4E
O	79	117	4F
P	80	120	50
Q	81	121	51
R	82	122	52
S	83	123	53
T	84	124	54
U	85	125	55
V	86	126	56
W	87	127	57
X	88	130	58
Y	89	131	59
Z	90	132	5A
[	91	133	5B
\	92	134	5C
]	93	135	5D
^	94	136	5E
_	95	137	5F
`	96	140	60
a	97	141	61
b	98	142	62
c	99	143	63
d	100	144	64
e	101	145	65
f	102	146	66
g	103	147	67
h	104	150	68
i	105	151	69
j	106	152	6A
k	107	153	6B
l	108	154	6C
m	109	155	6D
n	110	156	6E
o	111	157	6F
p	112	160	70
q	113	161	71
r	114	162	72
s	115	163	73
t	116	164	74
u	117	165	75
v	118	166	76
w	119	167	77
x	120	170	78
y	121	171	79
z	122	172	7A
{	123	173	7B
|	124	174	7C
}	125	175	7D
~	126	176	7E
■	127	177	7F

Table B-5. EBCDIC Character Codes

GRAPHIC	DEC	OCT	HEX
NUL	0	0	0
SOH	1	1	1
STX	2	2	2
ETX	3	3	3
PF	4	4	4
HT	5	5	5
	6	6	6
DEL	7	7	7
	8	10	8
	9	11	9
	10	12	A
VT	11	13	B
FF	12	14	C
CR	13	15	D
SO	14	16	E
SI	15	17	F
DLE	16	20	10
DC1	17	21	11
DC2	18	22	12
TM	19	23	13
RES	20	24	14
NL	21	25	15
BS	22	26	16
IL	23	27	17
CAN	24	30	18
EM	25	31	19
CC	26	32	1A
CU1	27	33	1B
IFS	28	34	1C
IGS	29	35	1D
IRS	30	36	1E
IUS	31	37	1F
DS	32	40	20
SOS	33	41	21
FS	34	42	22
	35	43	23
BYP	36	44	24
LF	37	45	25
ETB	38	46	26
ESC	39	47	27
	40	50	28
	41	51	29
SM	42	52	2A
CU2	43	53	2B
	44	54	2C
ENQ	45	55	2D
ACK	46	56	2E
BEL	47	57	2F
	48	60	30
	49	61	31
SYN	50	62	32
	51	63	33
PN	52	64	34
RS	53	65	35
UC	54	66	36
EOT	55	67	37
	56	70	38
	57	71	39
	58	72	3A
CU3	59	73	3B
DC4	60	74	3C
NAK	61	75	3D
	62	76	3E
SUB	63	77	3F

GRAPHIC	DEC	OCT	HEX
SP	64	100	40
	65	101	41
	66	102	42
	67	103	43
	68	104	44
	69	105	45
	70	106	46
	71	107	47
	72	110	48
	73	111	49
	74	112	4A
.	75	113	4B
<	76	114	4C
(	77	115	4D
+	78	116	4E
ßß	79	117	4F
&	80	120	50
	81	121	51
	82	122	52
	83	123	53
	84	124	54
	85	125	55
	86	126	56
	87	127	57
	88	130	58
	89	131	59
!	90	132	5A
$	91	133	5B
*	92	134	5C
)	93	135	5D
;	94	136	5E
¬	95	137	5F
-	96	140	60
/	97	141	61
	98	142	62
	99	143	63
	100	144	64
	101	145	65
	102	146	66
	103	147	67
	104	150	68
	105	151	69
¦	106	152	6A
,	107	153	6B
%	108	154	6C
_	109	155	6D
>	110	156	6E
?	111	157	6F
	112	160	70
	113	161	71
	114	162	72
	115	163	73
	116	164	74
	117	165	75
	118	166	76
	119	167	77
	120	170	78
	121	171	79
:	122	172	7A
#	123	173	7B
@	124	174	7C
'	125	175	7D
=	126	176	7E
"	127	177	7F

Table B-5. EBCDIC Character Codes (Continued)

GRAPHIC	DEC	OCT	HEX
	128	200	80
a	129	201	81
b	130	202	82
c	131	203	83
d	132	204	84
e	133	205	85
f	134	206	86
g	135	207	87
h	136	210	88
i	137	211	89
	138	212	8A
	139	213	8B
	140	214	8C
	141	215	8D
	142	216	8E
	143	217	8F
	144	220	90
j	145	221	91
k	146	222	92
l	147	223	93
m	148	224	94
n	149	225	95
o	150	226	96
p	151	227	97
q	152	230	98
r	153	231	99
	154	232	9A
	155	233	9B
	156	234	9C
	157	235	9D
	158	236	9E
	159	237	9F
	160	240	A0
~	161	241	A1
s	162	242	A2
t	163	243	A3
u	164	244	A4
v	165	245	A5
w	166	246	A6
x	167	247	A7
y	168	250	A8
z	169	251	A9
	170	252	AA
	171	253	AB
	172	254	AC
[	173	255	AD
	174	256	AE
	175	257	AF
	176	260	B0
	177	261	B1
	178	262	B2
	179	263	B3
	180	264	B4
	181	265	B5
	182	266	B6
	183	267	B7
	184	270	B8
	185	271	B9
	186	272	BA
	187	273	BB
	188	274	BC
]	189	275	BD
	190	276	BE
	191	277	BF

GRAPHIC	DEC	OCT	HEX
{	192	300	C0
A	193	301	C1
B	194	302	C2
C	195	303	C3
D	196	304	C4
E	197	305	C5
F	198	306	C6
G	199	307	C7
H	200	310	C8
I	201	311	C9
	202	312	CA
	203	313	CB
	204	314	CC
	205	315	CD
	206	316	CE
	207	317	CF
}	208	320	D0
J	209	321	D1
K	210	322	D2
L	211	323	D3
M	212	324	D4
N	213	325	D5
O	214	326	D6
P	215	327	D7
Q	216	330	D8
R	217	331	D9
	218	332	DA
	219	333	DB
	220	334	DC
	221	335	DD
	222	336	DE
	223	337	DF
\	224	340	E0
	225	341	E1
S	226	342	E2
T	227	343	E3
U	228	344	E4
V	229	345	E5
W	230	346	E6
X	231	347	E7
Y	232	350	E8
Z	233	351	E9
	234	352	EA
	235	353	EB
	236	354	EC
	237	355	ED
	238	356	EE
	239	357	EF
0	240	360	F0
1	241	361	F1
2	242	362	F2
3	243	363	F3
4	244	364	F4
5	245	365	F5
6	246	366	F6
7	247	367	F7
8	248	370	F8
9	249	371	F9
	250	372	FA
	251	373	FB
	252	374	FC
	253	375	FD
	254	376	FE
	255	377	FF

Programming Examples

APPENDIX C

This appendix presents several utility programs, written in BASIC, that you may find useful when implementing an applications program to drive an HP2622A Display Terminal or an HP 2623A Graphics Terminal.

FORMIO1

This program reads the content of display memory into the host computer and generates the sequence of PRINT statements necessary to recreate the data.

Figure C-1 shows a source listing of FORMIO1.

Once FORMIO1 is stored as a BASIC program file within the host computer, you use it as follows:

1. Switch the terminal to local mode, home the cursor, and clear the display.
2. Using the line drawing set, design the data entry form. When you are done, leave the cursor positioned at the start of a line anywhere BELOW the form.
3. Switch the terminal back to remote mode and, within the BASIC Interpreter, run FORMIO1.
4. FORMIO1 will ask you what starting line number and increment you wish used. For example, if you want the

```
>list
FORMIO1
   10 FILES *,*
   15 SYSTEM X1,"PURGE FDATA"
   20 SYSTEM X1,"BUILD FDATA;rec=-132,,f,ascii"
   30 SYSTEM X1,"FILE X=$stdin;rec=-256"
   40 ASSIGN "FDATA",1,A1
   50 ASSIGN "X",2,A1,WR
   60 DIM A$[255],A1$[6],C$[3]
   70 PRINT CTL(208),'27"F"'27"a";
   80 ENTER 255,X,A$
   90 CONVERT A$[8;3] TO R
   95 SYSTEM X1,'27";"
  100 PRINT "This program creates basic statements that define the"
  110 PRINT "FORM or other data in this terminal's memory.";LIN(3)
  120 INPUT "Starting statement number, increment  ?",A,B
  130 SYSTEM T9,'27+";"
  150 PRINT '27"h";
  160 PRINT #1;"scr";END
  170 FOR I=1 TO R
  180   PRINT '27"d";
  190   LINPUT #2;A$
  200   IF UPS$(A$[1,3])="RUN" THEN 500
  210   IF UPS$(A$[1,4])=">RUN" THEN 500
  220   CONVERT A TO A1$
  230   REM compensate for imbedded  "  marks
  240   C=-4
  250   IF C+5>LEN(A$) THEN 310
  260   C1=POS(A$[C+5],'34)
  270   IF NOT C1 THEN 310
  280   C=C1+C+4
  290   A$=A$[1,C]+"'34"+'34+A$[C+1]
  300   GOTO 250
  310   REM spaces >=7 are converted to direct cursor addresses
  320   FOR C=1 TO LEN(A$)
  330     IF A$[C,C+6]="       " THEN DO
  340       FOR C1=C+7 TO LEN(A$)
  350         IF A$[C1,C1]<>" " OR LEN(A$)=C1 THEN DO
  360           CONVERT C1-C TO C$
  370           A$[C]='27"&a"+DEB$(C$)+"C"+A$[C1]
  380           GOTO 310
  390         DOEND
  400       NEXT C1
  410     DOEND
  420   NEXT C
  430   REM output form record as a BASIC print statement
  440   PRINT #1;" "+A1$+" print ctl(208),&";END
  450   PRINT #1;'34+A$[1,LEN(A$) MIN 127];"&";END
  460   IF LEN(A$)<128 THEN PRINT #1;'34;END
  470   IF LEN(A$)>=128 THEN PRINT #1;A$[128]+'34;END
  480   A=A+B
  490 NEXT I
  500 PRINT '27"FNow type 'XEQ FDATA' then 'LIST'.";LIN(1)
  510 PRINT "These statements will reproduce your terminal's memory--"
  520 PRINT "modify, NAME, RENUM, and SAVE as you wish....."
  530 SYSTEM T9,'27+":"
  550 END
```

Figure C-1. FORMIO1 Source Listing

PRINT statements numbered starting with 10 and proceeding in increments of 10, type "10, 10" and then press RETURN.

5. FORMIO1 homes the cursor and reads each line until the cursor reaches the line containing the RUN command.
6. Enter "XEQ FDATA" and then press RETURN. FORMIO1 is now replaced in the BASIC Interpreter work space by the PRINT statements which, when executed, will recreate the data that was read from display memory.

At this point you may do with the PRINT statements as you like (LIST them, RUN them, NAME and SAVE them, add more statements to them, and so forth).

LRGLINE

This program accepts characters entered through the keyboard and replaces them on the screen with large characters created by the line drawing set.

Figure C-2 shows a source listing of LRGLINE.

Once LRGLINE is stored as a BASIC program file within the host computer, you use it as follows:

1. Within the BASIC Interpreter, run LRGLINE.
2. Move the cursor with the space bar, then type the desired string of characters and then press RETURN. Starting at the cursor position, LRGLINE recreates the string using the line drawing set and then leaves the cursor positioned at the left margin in the third line below the large character string.
3. Repeat step #2 for as many character strings as desired.
4. To stop the program, enter a control-Y and then type "ABORT".

With the terminal in local mode, use the edit keys (insert character, delete character, insert line, and delete line) to reposition the characters as desired and then, in remote mode, use the FORMIO1 program to generate the PRINT statements necessary to recreate the large character string(s).

```
LRGLINE
   10 PRINT LIN(4)
   20 LINPUT L1$
   30 IF LEN(L1$)=0 THEN GOTO 150
   40 L1$=UPS$(L1$)
   50 J$=DEB$(L1$)
   60 IF LEN(J$)=0 THEN GOTO 150
   70 I=1
   80 IF L1$[I,I]<>" " THEN GOTO 110
   90 I=I+1
  100 GOTO 80
  110 PRINT CTL(208),'27"&a-1r";I-1;"C"+'27"4";
  120 L1$=DEB$(L1$)
  130 IF 6*LEN(L1$)>78-I THEN L1$=L1$[1,INT((78-I)/6)]
  140 GOSUB 9000
  150 PRINT CTL(208),'10'10'10'10'10'10'10'27"&a0C";
  160 PRINT CTL(208),'27"9";
  170 GOTO 20
 9000 REM SUBROUTINE TO PRINT STRING IN L1$ IN LINE DRAWING SET
 9010 REM USES VARIABLES L1$ THRU L4$,L1 THRU L3
 9020 DIM L1$[80],L2$[11],L3$[128,75],L4$[1]
 9025 INTEGER L4
 9030 IF UND(L1)=1 THEN DO
 9040   L3$[32]="      \      \      \      \      \"
 9050   L3$[33]="  c  \  c  \  c  \      \  x  \"
 9060   L3$[34]=" c c \      \      \      \"
 9070   L3$[35]=" c c \zczcz\ c c \zczcz\ c c \"
 9100   L3$[39]="  c  \ "'27"&dBz"'27"&d@    \ '    \      \      \"
 9160   L3$[44]="      \      \      \  c  \ "'27"&dBz"'27"&d@  \"
 9180   L3$[46]="      \      \      \      \  c  \"
 9200   L3$[48]="x"'27"&dBzzz"'27"&d@x\c    c\c    c\c    c\"'27"&dBz"'27&
    "&d@xxx"'27"&dBz\"'27"&dBz\"
 9210   L3$[49]=" "'27"&dBz"'27"&d@c  \  c  \  c  \  c  \xxcxx\"
 9220   L3$[50]="c"'27"&dBzzz"'27"&d@c\     c\c"'27"&dBzzzz\c     \cxxxc\"
 9230   L3$[51]="c"'27"&dBzzz"'27"&d@c\     c\ "'27"&dBzzz"'27&
    "&d@c\     c\cxxxc\"
 9240   L3$[52]="c    c\c    c\"'27"&dBzzzz"'27"&d@c\     c\     c\"
 9250   L3$[53]="c"'27"&dBzzzz\c     \"'27"&dBzzzz"'27"&d@x\     c\cxxxc\"
 9260   L3$[54]="c"'27"&dBzzz"'27"&d@c\c     \c"'27"&dBzzz"'27&
    "&d@x\c    c\cxxxc\"
 9270   L3$[55]="c"'27"&dBzzz"'27"&d@c\     c\     c\     c\     c\"
 9280   L3$[56]="x"'27"&dBzzz"'27"&d@x\c    c\z"'27"&dBzzz"'27&
    "&d@z\c    c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
 9290   L3$[57]="x"'27"&dBzzz"'27"&d@x\c    c\"'27"&dBxzzz"'27&
    "&d@c\     c\     c\"
 9300   L3$[58]="      \  c  \      \  c  \      \"
 9310   L3$[59]="      \  c  \      \  c  \ "'27"&dBzx"'27"&d@  \"
 9360   L3$[66]="c"'27"&dBzzz"'27"&d@x\c    c\c"'27"&dBzzz"'27&
    "&d@z\c    c\cxxx"'27"&dBz\"
 9370   L3$[65]="x"'27"&dBzzz"'27"&d@x\c    c\cxxxc\c    c\c    c\"
```

Figure C-2. LRGLINE Source Listing

```
9380   L3$[67]="x"'27"&dBzzz"'27"&d@x\c    "'27"&dBx\c     \c    x\"'27&
    "&dBz"'27"&d@xxx"'27"&dBz\"
9390   L3$[68]="c"'27"&dBzzz"'27"&d@x\c    c\c    c\c    c\cxxx"'27"&dBz\"
9400   L3$[69]="c"'27"&dBzzzz\c     \c"'27"&dBzz"'27"&d@  \c     \cxxxx\"
9410   L3$[70]="c"'27"&dBzzzz\c     \cxxx \c     \c     \"
9420   L3$[71]="x"'27"&dBzzz"'27"&d@x\c    "'27"&dBx\c xxx\c    c\"'27&
    "&dBz"'27"&d@xxx"'27"&dBz\"
9430   L3$[72]="c    c\c    c\c"'27"&dBzzz"'27"&d@c\c    c\c    c\"
9440   L3$[73]="'27"&dBzz"'27"&d@c"'27"&dBzz\  "'27&
    "&d@c  \  c  \  c  \xxcxx\"
9450   L3$[74]="     c\     c\     c\x    c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
9460   L3$[75]="c    c\c   c"'27"&dBx\cccz \c  c \c  cc\"
9470   L3$[76]="c     \c     \c     \c     \cxxxx\"
9480   L3$[77]="cczcc\c c c\c "'27"&dBz"'27"&d@ c\c    c\c    c\"
9490   L3$[78]="c    c\cc   c\c c c\c   cc\c    c\"
9500   L3$[79]="x"'27"&dBzzz"'27"&d@x\c    c\c    c\c    c\"'27"&dBz"'27&
    "&d@xxx"'27"&dBz\"
9510   L3$[80]="c"'27"&dBzzz"'27"&d@x\c    c\c"'27"&dBzzz"'27&
    "&dBx\c     \c     \"
9520   L3$[81]="x"'27"&dBzzz"'27"&d@x\c    c\c    c\c "'27"&dBz"'27&
    "&d@zc\"'27"&dBz"'27"&d@xxx"'27"&dBx\"
9530   L3$[82]="c"'27"&dBzzz"'27"&d@x\c    c\cxxx"'27"&dBz\c  c \c   cc\"
9540   L3$[83]="x"'27"&dBzzz"'27"&d@x\c     \"'27"&dBz"'27&
    "&d@xxxz\     c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
9550   L3$[84]="'27"&dBzz"'27"&d@c"'27"&dBzz\  "'27&
    "&d@c  \  c  \  c  \  c  \"
9560   L3$[85]="c    c\c    c\c    c\c    c\"'27"&dBz"'27"&d@xxx"'27"&dBz\"
9570   L3$[86]="c    c\c    c\c    c\"'27"&dBx"'27"&d@c c"'27"&dBx\ "'27&
    "&dBx"'27"&d@c"'27"&dBx"'27"&d@ \"
9580   L3$[87]="c    c\c    c\c x c\c c c\cc"'27"&dBx"'27"&d@cc\"
9590   L3$[88]="c    c\ c c \  c  \ c c \c    c\"
9600   L3$[89]="c    c\c    c\"'27"&dBz"'27"&d@c c"'27"&dBz\  c  \  c  \"
9610   L3$[90]="'27"&dBzzzz"'27"&d@c\    c \  c  \ c    \cxxxx\"
9800 DOEND
9810 PRINT CTL(208);'27")B";
9820 FOR L1=1 TO LEN(L1$)
9830   PRINT CTL(208),'27"4";
9840   L3=1
9850   L2=NUM(L1$[L1,L1])
9851   IF POS(L3$[L2,1,75],"\")=0 THEN L2=32
9860   FOR I=1 TO 5
9870     L4=POS(L3$[L2,L3,75],"\")
9875     IF L4=0 THEN RETURN
9880     PRINT CTL(208);'14+L3$[L2,L3,L3+L4-2]+'15+'27"&d@ ";
9890     L3=L4+L3
9900     PRINT CTL(208);'15'13'10;
9910   NEXT I
9920   PRINT CTL(208),'27"&a-5r+6C"+'27"4";
9930 NEXT L1
9940 RETURN
```

Figure C-2. LRGLINE Source Listing (Continued)

2622A/2623A REFERENCE MANUAL
Part No. 02622-90008 **Printed: 7/82**

Your Comments, Please . . . . .

Your comments assist us in improving the usefulness of our publications. Comments are an important part of the inputs used in preparing updates to manuals.

Please complete the questionnaire, fold it up and return it to us. Feel free to mark more than one box to a question and to make any additional comments. If you prefer not to give us your name just leave the last part, name and address, blank. All comments and suggestions become the property of HP.

Thank you for your help.

1. Did you have any difficulty in understanding or applying the material presented in this manual?

☐ None ☐ Minimal Difficulty ☐ Difficulty ☐ Considerable Difficulty

If so:

fold — *fold*

a. What were the "difficult" areas?

☐ Indexing? ☐ Depth of coverage? ☐ Examples?
☐ Organization? ☐ Omitted information? ☐ Other (Please explain)

b. How could we clarify these areas? ______________________________

____

2. What errors or misleading information have you found in the manual? (*Please give page numbers.)*

____

3. What was your level of programming knowledge before you started using this manual?

☐ None ☐ Beginner ☐ Intermediate ☐ Expert

4. What is your major application of the terminal?

☐ Business Adminstration ☐ Interfacing ☐ Medicine
☐ General Computation ☐ Education What kind?
☐ Engineering ☐ Data Entry ☐ Other
What kind? ☐ Control

fold — *fold*

5. What best describes how you use the terminal?

☐ Run programs written by others ☐ Do applications programming ☐ Hobbyist
☐ Write simple programs ☐ Write complex programs ☐ Other
☐ Do simple calculations

6. General Comments: ______________________________

____

____

____

Name: ______________________________

Address: ______________________________

Thank you for your help. No postage necessary if mailed in the U.S.A.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 506 CUPERTINO, CA., U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard
Attn: Marketing Publications
974 East Arques Avenue
Sunnyvale, California, U.S.A., 94086

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X